BASIC-OPAS

Jan Lundgren
Sören Thornell

otava

BASIC II

Jan Lundgren
Sören Thornell

DKA
61.39

Kotkan
kaupunginkirjasto

2

HELSINGISSÄ KUSTANNUSOSAKEYHTIÖ OTAVA

Esipuhe

Meidän tietokoneistetussa yhteiskunnassamme ovat korvaamattoman tärkeitä perustiedot siitä, miten tietokoneita käytetään ja miten niitä ohjelmoidaan. Tehokkaat tietokoneet, joita nykyään on saatavissa halvalla, tarjoavat yhä useammille mahdollisuuden hankkia ja käyttää hyväksi näitä tietoja.

BASIC II -opas selittää helpolla tavalla, miten nykyaikaista tietokonejärjestelmää ohjelmoidaan ja käytetään. Samalla painotetaan sitä, miten tärkeätä on oppia hyviä ohjelmointitapoja ja laatia ohjelmia käyttäjän tarpeiden mukaan.

Kirjan neljä ensimmäistä lukua antavat yksinkertaiseen tietojenkäsittelyyn tarvittavat perustiedot. Värin käyttömahdollisuuksia kuvaruudulle tulostettaessa kuvataan luvussa 5.

Kaikelle tietojenkäsittelylle oleellista tiedostojen käsittelyä kuvataan perusteellisesti luvuissa 6 ja 8. Luvussa 7 tarkastellaan ohjelmien rakennetta. Loput luvut sisältävät mm. yksinkertaista sanojenkäsittelyä, matemaattisia sovellutuksia ja erilaisia graafisen esityksen muotoja.

Jokaisessa luvussa on suuri määrä esimerkkiohjelmia, jotka kuvaavat käytännöllistä tietojen käsittelyä. Esimerkkiohjelmat, kuten myös lukuisat harjoitustehtävät ratkaisuineen, helpottavat kirjan käyttöä yksinopiskelussa. Kirjaa voidaan käyttää myös lukioissa, ammattikouluissa sekä kansalaisopistoissa.

Kirjan esimerkkiohjelmat ja harjoitustehtävien ratkaisut ovat saatavissa tietolevyllä.

Me kiitämme kaikkia, jotka eri tavoin ovat auttaneet tämän kirjan syntymistä. Ne sisältöä ja muotoa koskevat näkökohdat, joita Margit E. Lundgren esitti työn aikana, ovat olleet hyvin arvokkaita.

Linköping, huhtikuussa 1982

Jan Lundgren

Sören Thornell

1. painos

Suomentajat: Usko Lahti, Harri Lehtimäki
Piirrokset: Marianne Lundin, Tuovi Laine
Valokuvat: WT-KONSULT, EMMDATA, Raimo Immonen

© 1982 Jan Lundgren, Sören Thornell
© 1983 suomenkielinen laitos Usko Lahti, Harri Lehtimäki ja Kustannusosakeyhtiö Otava

Kustannusosakeyhtiö Otavan painolaitokset Keuruu 1984

ISBN 951-1-07316-8

Sisällys

1 Mikä on tietokone 7
2 Mikä on tietokoneohjelma 15
3 Tietokone suorittaa valintoja 27
4 Tietokone laskee 49
5 Värien käyttö 71
6 Miten valmiita ohjelmia talletetaan 87
7 Ohjelman rakenne 103
8 Tiedostojen käsittely 123
9 Sanojenkäsittelyn perusteita 151
10 Matemaattisia laskuja 169
11 Graafisia mahdollisuuksia 187
12 Harjoitustehtävien vastaukset 207
13 Asiahakemisto ja viitekirjallisuus 233
14 ABC 800 ohjelmointikortti

Suomentajan esipuhe

Tämä nyt kädessäsi oleva kirja on johdatus erinomaisessa ruotsalaisessa ABC 800 -tietokoneessa käytettävän BASIC II -ohjelmointikielen perusteisiin. BASIC II -ohjelmointikieli on laajennettu versio yleisesti käytössä olevasta BASIC-ohjelmointikielestä. Uusia piirteitä ovat mm. sellaiset ohjelmointityötä helpottavat rakenteet kuten WHILE-silmukat ja funktioaliohjelmien määrittelyt.

Lukuisat esimerkkiohjelmat tekevät kirjasta helppotajuisen ja mielenkiintoisen. Kääntäessämme esimerkkiohjelmia ja harjoitustehtäviä olemme pyrkineet soveltamaan ohjelmia Suomen oloja vastaamaan. Paljon varsinaisia muutoksia ei ole silti tarvinnut tehdä alkuperäisiin ohjelmiin, sillä ohjelmat olivat varsin yleisiä ja hyviä. Käännöstyössäkin saatoimme tyytyä lähes sanatarkkaan suomentamiseen, sillä alkuperäisteos on selkeästi ja havainnollisesti kirjoitettu.

Haluamme kiittää Electrolux-Facitia, Luxor Dataa sekä erityisesti Atk-palvelu Martti Rauniota saamastamme henkisestä tuesta ja käännöstyön ajaksi käyttöömme saamistamme tietokoneista ja tekstinkäsittelyohjelmista. Ilman mainiota SANA-ohjelmaa kirjoitustyö olisi ollut paljon vaivalloisempaa. Kiitämme myös Kustannusosakeyhtiö Otavaa kunnollisen ohjelmointioppaan kustantamisesta.

Helsingissä, maaliskuussa 1983

Usko Lahti
Harri Lehtimäki

1

Mikä on tietokone?

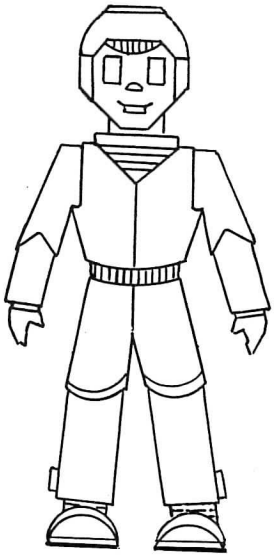
Tietokoneet arkikäytössämme

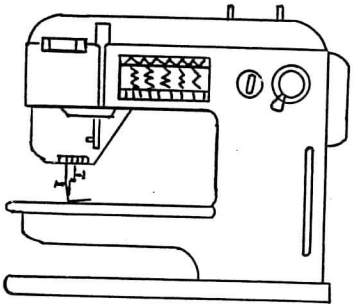
Jokapäiväisessä elämässämme olemme kaikki tavalla tai toisella tekemisissä tietokoneiden kanssa.

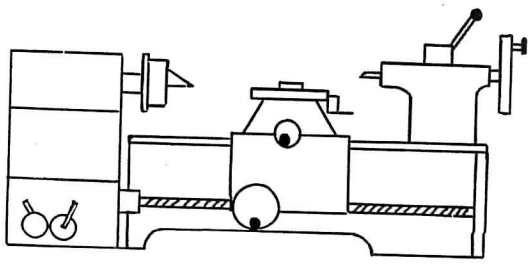
OSA 11 SEKTION
RIVI 14 RAD
PAIKKA 6 PLATS
APUKATSOMO
NORM TMM0813
73.00
APU
VM I FRIIDROTT
YLEISURHEILUN MM 83
IAAF WORLD CHAMPIONSHIPS
OLYMPIASTADION
* * *
LA 13.08.83 KLO 17.00
KATSOMO
OSA 11
KL 3PBO
RIVI 14
N 73.00
PAIKKA 6
TARKISTAKAA LIPPUNNE, OLKAA HYVÄ!
KONTROLLERA ERA BILJETTER, VAR SÅ VÄNLIG!
Nº 063783

Kun otamme rahaa pankkiautomaatista tai ostamme junalipun, koemme tietokoneet luonnollisiksi apuvälineiksi. Mainittuihin tarkoituksiin käytettävillä tietokoneilla on hyvin suuri kapasiteetti ja niitä voivat käyttää useat henkilöt samanaikaisesti.

Elektroniikan kehityksen ansiosta voidaan pieniä ja tehokkaita tietokoneita (mikrotietokoneita) valmistaa halvalla.







Tästä johtuu, että nykyisin tietokoneita löytyy mitä erilaisimmista tuotteista: leikkikaluista, ompelukoneista, sorveista jne.

Monipuolinen tietokone

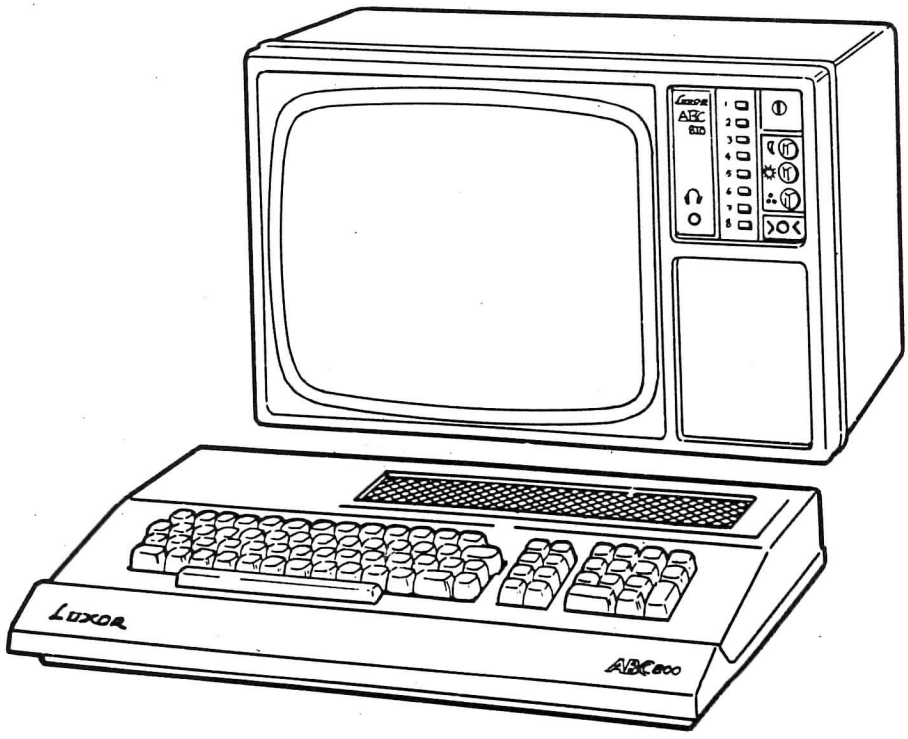
Yleensä on välttämätöntä, että tietokonetta voidaan soveltaa erilaisissa tehtävissä ja toiminnoissa. Esimerkkeinä mainittakoon koneen ohjaus tehtaassa, kirjeen kirjoittaminen konttorissa ja matemaattisen laskun suorittaminen koulussa.

Tietokoneita, joita voidaan käyttää eri tarkoituksiin, on ollut olemassa jo pitkään. Viime vuosien kehitys on kuitenkin johtanut siihen, että tällaisia yleisiä tietokonejärjestelmiä on nyt saatavissa hyvin halvalla. Koko on pienentynyt ratkaisevasti: täydellinen tietokonejärjestelmä sopii tavalliselle kirjoituspöydälle.

Ruotsalaiset tietokoneet ABC 80 ja ABC 800 ovat tällaisia yleisiä tietokonejärjestelmiä.

ABC 800 -tietokone

ABC 800 -tietokone koostuu perusmuodossaan kahdesta yksiköstä: näppäinpöydästä ja TV-monitorista.

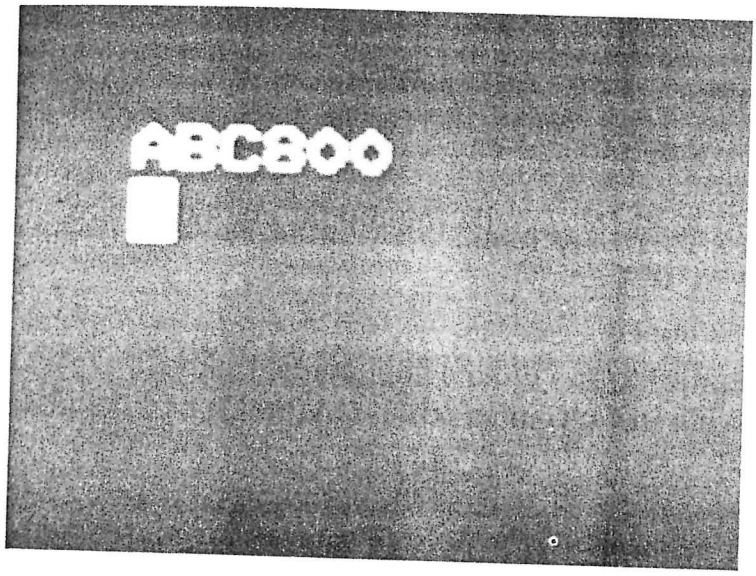


Näppäinpöytä sisältää integroituja piirejä, jotka yhdessä muodostavat tietokoneen perusyksikön. Käyttäjä antaa tietokoneelle tehtäviä näppäimistön välityksellä. Tietokone puolestaan esittää informaatiota käyttäjälle kuvaruudun kautta.

Lue käyttöohje

Käyttöohjeessa on seikkaperäinen kuvaus siitä, kuinka yksiköt kytketään yhteen. Siellä on myös selvitetty kuinka kuvaruutu säädetään.

Kun yksiköt on kytketty, voidaan tietokone käynnistää verkkokytkimen painalluksella. Seuraava teksti on kuvaruudulla, kun tietokone on valmis käyttöön.

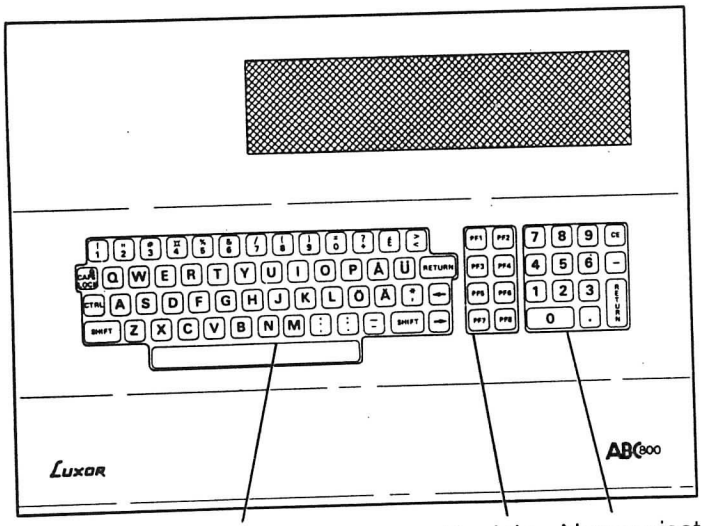


Kohdistimen tehtävänä on antaa se paikka kuvaruudulta, johon seuraava merkki ilmaantuu. Siirron jälkeen vilkuttaa kohdistin noin seitsemän sekunnin ajan. Kohdistimen uusi asema on tällä tavoin helpompi havaita.

Syöttö näppäimistön kautta

Näppäinpöydän näppäimet on jaettu kolmeen eri kenttään. Tavallisesti käytetään vasenta ryhmää, joka sisältää kirjain- ja numeromerkit (aakkosnumeeriset näppäimet) kuten tavallisessa kirjoituskoneessa.

Tämän mukaan esim. näppäintä [SHIFT] yhdistettynä ylimmän rivin näppäimiin käytetään saamaan aikaan merkit ! ” # ⌖ jne.

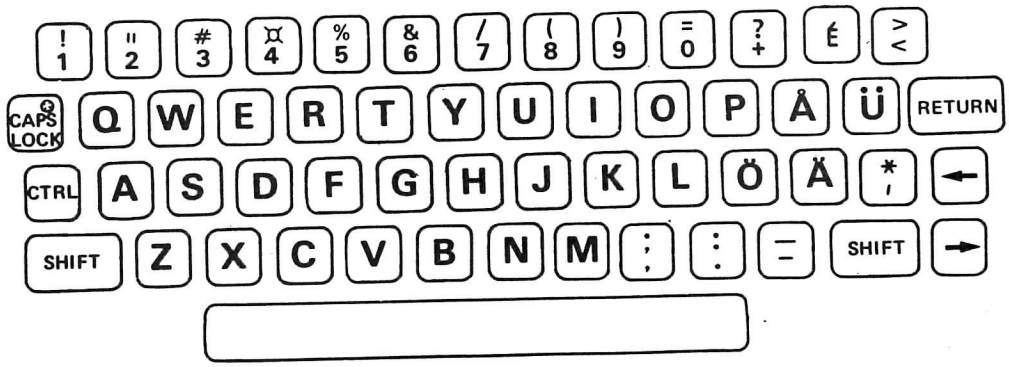


Kokeile näppäimistöä kirjoittamalla muutamia merkkejä esim.

print

Painamalla näppäintä CAPS LOCK voidaan näppäimistö ''lukita'' (engl. LOCK), niin että vain isot kirjaimet (engl. CAPitalS) ovat käytössä.

Uusi painallus näppäimeen CAPS LOCK poistaa suurten kirjainten lukituksen. Kuvassa väritetyt näppäimet ovat näppäimen CAPS LOCK vaikutuspiirissä.

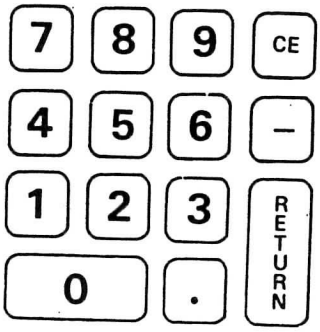


Kokeile tätä painamalla näppäintä CAPS LOCK ja kirjoittamalla sen jälkeen muutamia merkkejä, esim.

PRINT

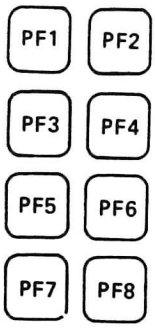
Erityisnäppäinten kentät

Sovellutuksissa, joissa tietokoneelle syötetään useita lukuarvoja, käytetään menestyksellisesti oikeanpuoleista näppäinkenttää (numeerisia näppäimiä).



Nopea syöttö helpottuu näppäinten sijoittelun ansiosta.

Funktionäppäinten kenttä on keskellä näppäinpöytää. Funktionäppäimillä on kokonaan toisenlainen luonne kuin muilla näppäimillä.



Funktionäppäimiä käsitellään tuonnempana.

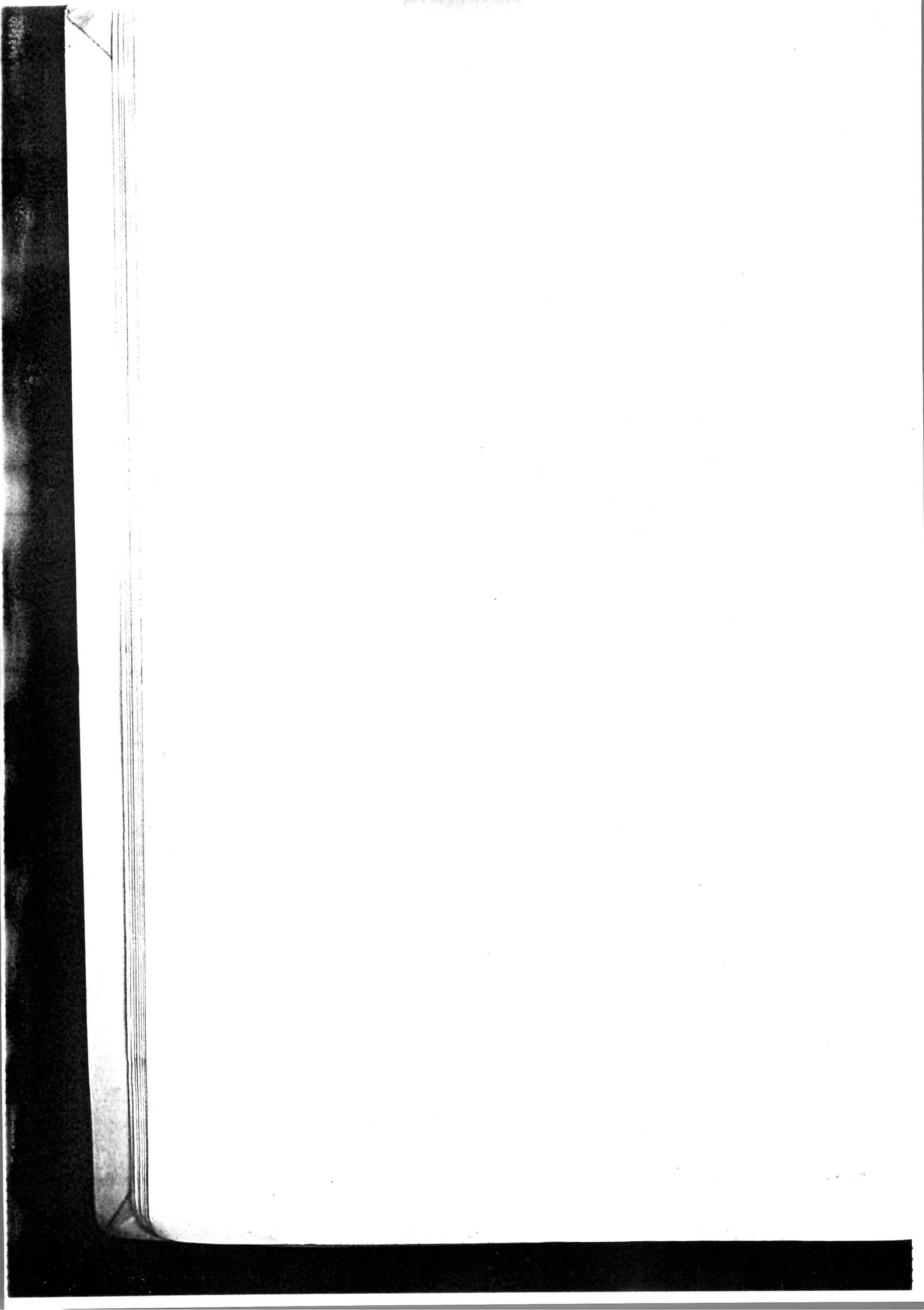
Tärkeä ohjelma

Voidakseen suorittaa tehtävän täytyy tietokoneen seurata työkuvausta, ohjelmaa. Ohjelman muotoilu on ratkaisevan tärkeä tietokoneen toiminnan kannalta. Tämä kirja antaa Sinulle perusteet ohjelmien kirjoittamiseksi ABC 800 -tietokoneelle.

Moniin sovellutuksiin on saatavissa valmiita standardiohjelmia. Myös käyttäessäsi tällaisia ohjelmia on Sinulle suurta hyötyä siitä, että tunnet ohjelmoinnin perusteet.

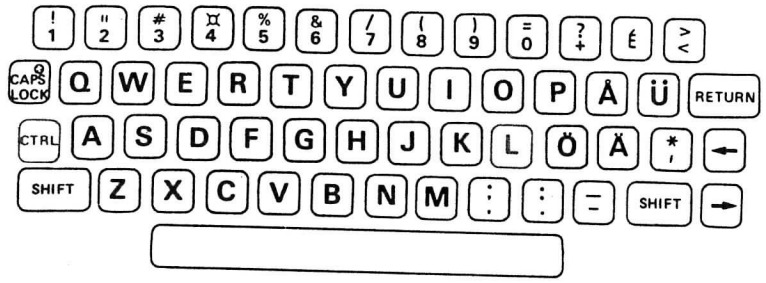
2

Mikä on tietokone-ohjelma?



Kuvaruudun tyhjentäminen

Luvun 1 alustavien kokeilujen jälkeen on nyt sopivaa ottaa teksti pois kuvaruudulta.



Pidä näppäin CTRL alaspainettuna ja paina samaan aikaan näppäintä L . Tästä seuraa, kuten tulet näkemään, että kuvaruutu tyhjenee ja kohdistin siirtyy kuvaruudun vasempaan yläkulmaan.

Tietokoneen ensimmäinen tehtävä

Tietokoneen ensimmäisenä tehtävänä on kirjoittaa nimesi kuvaruudulle. Tehtävät, jotka tietokoneen tulee suorittaa, täytyy kirjoittaa (englanninkielisinä) käskyinä. Tässä tapauksessa KIRJOITA-sanaa vastaa PRINT.

Näppäile nyt PRINT ja sen perään lainausmerkeissä nimesi.

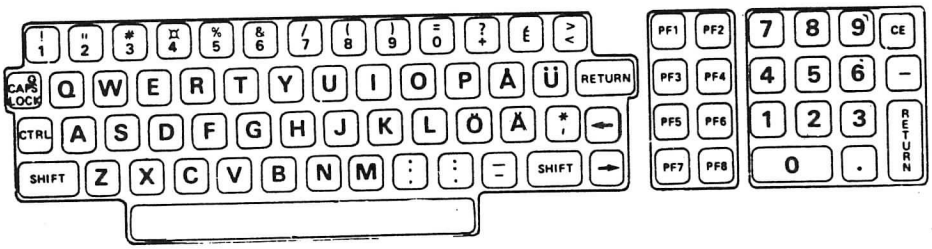
PRINT "MAIJA SUOMALAINEN" RETURN

Lopeta rivi painamalla näppäintä RETURN Mitä tapahtuu? Tietokone kirjoittaa nimesi kuvaruudulle, esim. seuraavalla tavalla:

Merkintä ABC800 kolmannella rivillä on tietokoneen ilmoitus siitä, että tehtävä on suoritettu. Jatkossa käytämme yksinomaan suuria kirjaimia. Tämä saadaan helpoiten aikaan painamalla alas näppäin [CAPS LOCK] niin, että sisäänrakennettu valodiodi syttyy.

Virheellisen syötön korjaus

Poistonäppäimen [←] avulla voit helposti poistaa väärin syötetyn merkin.



Jokainen poistonäppäimen painallus siirtää osoitinta yhden askeleen vasemmalle ja poistaa samalla merkin. Muiden näppäinten tapaan poistonäppäin toistuu itsestään, kun sitä pidetään alaspainettuna runsaan sekunnin ajan.

Ole ystävällinen ja kirjoita mielivaltainen määrä merkkejä ja kokeile poistonäppäimen toimintaa.

Jos syystä tai toisesta haluat poistaa kokonaisen rivin, voit käyttää näppäintä [CE] (engl. Clear Entry).

Ensimmäinen ohjelmasi

Miten menettelet, ellei tietokoneen suoritettavaksi tarkoitettuja tehtäviä voi kirjoittaa yhdelle riville? Niin, varustat jokaisen rivin rivinumerolla. Jos esimerkiksi haluat kirjoittaa sekä nimen että puhelinnumeron kuvaruudulle, voivat käskysi tietokoneelle näyttää seuraavalta:

```
10 PRINT "MAIJA SUOMALAINEN"    RETURN
20 PRINT "PUH 90-111 111"       RETURN
```

Entiseen tapaan lopetat jokaisen rivin painamalla näppäintä RETURN . Näitä kahta riviä, rivinumerolla varustettuina, kutsumme ohjelmaksi. Kun ohjelman molemmat rivit on syötetty, on aika ajaa ohjelma. Kirjoita tietokoneelle komento, joka aikaansaa tämän.

```
RUN    RETURN
```

Huomaa, että myös tässä lopetat painamalla näppäintä RETURN . Ajon jälkeen näyttää kuvaruutu seuraavalta:

Ohjelma tallentuu muistiin

Syötetty ohjelma on tietokoneen muistissa. Tämä merkitsee sitä, että ajokomentoa voidaan toistaa miten monta kertaa tahansa.

Kokeile tätä kirjoittamalla:

```
RUN    RETURN
```

Kuten huomaat, antaa myös tämä ajo saman tulostuksen. Voit tyhjentää kuvaruudun uutta ajoa varten näppäimillä CTRL L tavalliseen tapaan. Kokeile tätä!

Muistista kuvaruudulle

Poistamisen jälkeen ei ohjelma ole kuvaruudulla. Sitävastoin on ohjelma tallella tietokoneen muistissa, josta se saadaan uudelleen näkyviin komennolla:

LIST [RETURN]

Ohjelman täydentäminen

Ohjelmaa, jonka nyt näet kuvaruudulla, voidaan helposti täydentää mielivaltaisen monella ohjelmarivillä. Lisää esimerkiksi kaksi riviä täydellistä osoitettasi varten:

```
12 PRINT "UUDENMAANKATU 123"        [RETURN]
14 PRINT "00120 HELSINKI 12"        [RETURN]
```

Tyhjennä nyt kuvaruutu näppäimillä [CTRL] [L] ja tulosta sen jälkeen täydellinen ohjelma komennolla:

LIST [RETURN]

Tietokone järjestelee ohjelmarivit

Kuten näet, tulostaa tietokone ohjelmarivit oikeassa järjestyksessä. Vastaavalla tavalla suorittaa tietokone ajossa ohjelmarivit nousevassa numerojärjestyksessä. Kokeile tätä antamalla komento:

RUN RETURN

Tietokone kirjoittaa nimesi, osoitteesi ja puhelinnumerosi oikeassa järjestyksessä.

Ohjelman muuttaminen

Jos tahdot muuttaa ohjelmariviä, voit yksinkertaisesti kirjoittaa rivin uudelleen. Samalla rivinumerolla kirjoitettu uusi rivi tuhoaa automaattisesti entisen.

Jos tahdot kokonaan poistaa rivin tietokoneen muistista, kirjoitat vain rivinumeron uudelleen. Esimerkiksi rivi 20 tuhoutuu, kun kirjoitat:

20 RETURN

Tarkista, että rivi 20 on tuhoutunut, tulostamalla uusi ohjelmalistaus kuvaruudulle komennolla LIST.

Tietokoneen muistin pyyhkiminen

Tietokoneen muistissa oleva ohjelma tuhoutuu täydellisesti komennolla:

NEW RETURN

Ohjelma ei enää ole tallella tietokoneessa. Tarkista tämä komennolla:

LIST RETURN

Vain merkintä

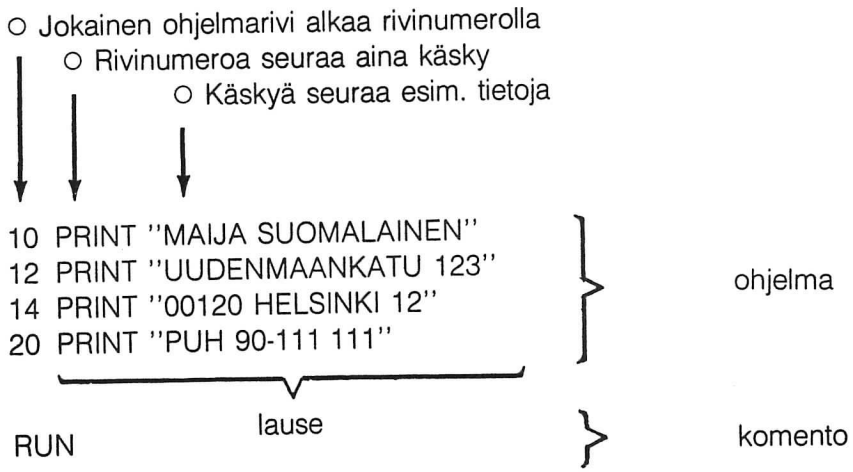
ABC 800

näkyy kuvaruudulla ja tietokoneen muisti on niin muodoin tyhjentynyt.

Ohjelmointikieli BASIC

Kun tietokoneita ohjelmoidaan, on ehdottomana vaatimuksena tiettyjen sääntöjen noudattaminen. BASIC-ohjelmointikielellä kirjoitettu ohjelma koostuu tietystä määrästä numeroituja ohjelmarivejä (BASIC = Beginners' All-purpose Symbolic Instruction Code).

Erilaisilla komennoilla, esim. RUN, LIST ja NEW, ilmoitetaan kuinka tietokoneen tulee käsitellä ohjelmaa.



Komentoa ei varusteta rivinumerolla.

Harjoitustehtävät

Jokaisen luvun lopussa on harjoitustehtäviä jotka Sinun tulee yrittää ratkaista. Ne harjoitukset, jotka ovat olennaisen tärkeitä tulevaa opetusta ajatellen, on merkitty tähdellä.

Kirjan eri luvuissa käsitellään BASIC-ohjelmointikieltä. Sen lisäksi kirjan viimeisillä sivuilla on täydellinen aakkosellinen kokoelma mm. käskyistä ja komennoista (kopio ABC 800 ohjelmointikortista).

Kirjan loppupuolella on täydelliset ratkaisut kaikkiin harjoitustehtäviin.

Yhteenveto

Jokainen luku lopetetaan jatkossa ohjelmointikortin otteella. Näin selviää mitkä uudet käskyt ja komennot luvussa on otettu käyttöön.

Käskyjä

PRINT ... (vaihtoehtoisesti ; ...)
Tulostaa kuvaruutuun. Käskyä voi seurata lauseke.

```
20 PRINT ''RIITTA''
30 PRINT A, B, C     (Tulostus sarakkeisiin)
40 PRINT ''ALA = ''; Y     (Tiheä tulostus)
50 ; ''KAISA''          (PRINT lyhennetty)
```

Komentoja

LIST ...
Tulostaa ABC 800:ssa olevan ohjelman kuvaruutuun. Seuraavat muunnokset ovat olemassa:

LIST	(kaikki rivit)
LIST 30	(vain yksi rivi)
LIST 30—60	(rivit kahden rivinumeron välillä)
LIST 60—	(rivit alkaen tietystä rivinumerosta)

NEW (vaihtoehtoisesti SCR)
Pyyhkii ohjelman ABC 800:n muistista.

RUN
Ajaa ABC 800:n muistissa olevan ohjelman.

Harjoitustehtäviä

Huom! Muista pyyhkiä tyhjäksi tietokoneen muisti komennolla NEW jokaisen suoritetun harjoituksen jälkeen.

* 2.1 Koeaja seuraava ohjelmaesimerkki.

```
10 PRINT "TÄMÄ"
20 PRINT "ON ENSIMMÄINEN"
30 PRINT "YRITYS"
```

* 2.2 Kuinka tulostus muuttuu (tehtävään 2.1 verrattuna), kun seuraava ohjelma ajetaan?

```
5 PRINT
10 PRINT "TÄMÄ"
20 PRINT "ON ENSIMMÄINEN"
30 PRINT "YRITYS"
40 PRINT
```

* 2.3 Muuta tehtävää 2.2 niin, että saadaan seuraava tulostus.

```
TÄMÄ
ON TOINEN
YRITYS
```

* 2.4 Laadi ohjelma, joka antaa seuraavan tulostuksen:

```
**********
*        *
*  DIPLOMI  *
*        *
* MAIJA SUOMALAINEN *
*        *
**********
```

* 2.5 Koeaja seuraava ohjelmaesimerkki ja tutki pilkun vaikutusta PRINT-lauseessa.

```
10 PRINT "TUOTE", "LUKUMÄÄRÄ", "HINTA"
20 PRINT "KAULAKETJU", "3", "225,—"
```

* 2.6 Täydennä tehtävän 2.5 ohjelmaa siten, että saat seuraavan tulostuksen:

TUOTE	LUKUMÄÄRÄ	HINTA
KAULAKETJU	3	225,—
PIIKKARIT	7	340,—
ONKIMATO	12	2,—

* 2.7 Koeaja seuraava ohjelma ja tutki puolipisteen vaikutusta PRINT-lauseessa.

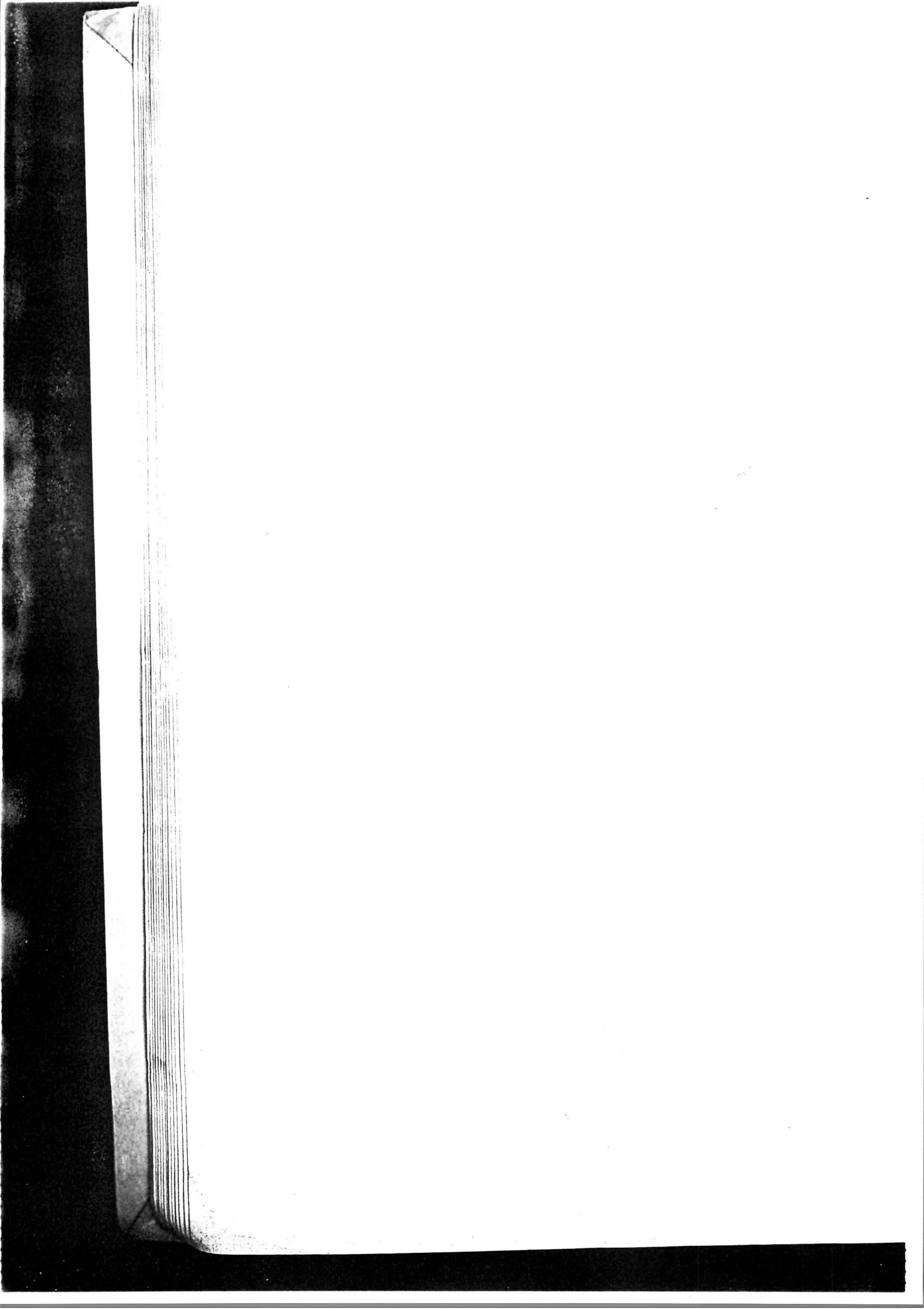
```
10 PRINT "KAKSI";
20 PRINT "TOISTA" " OPETUSLASTA"
```

* 2.8 Mihin sarakkeeseen sijoittuu tulostus, kun seuraava ohjelma ajetaan?

```
10 PRINT ,"O T S I K K O"
20 PRINT ,"= = = = = = = ="
```

* 2.9 Tämä ohjelmaesimerkki osoittaa, kuinka käsky PRINT voidaan lyhentää puolipisteellä (;)

```
10 ;
20 ; "** SELITYS ***"
30 ;
40 ; "Tämä ohjelma näyttää kuinka PRINT"
50 ; "voidaan lyhentää käyttämällä puo-"
60 ; "lipistettä (;)"
70 ;
```



3

Ti[illegible]kone [illegible]ittaa v[illegible]ntoja

Nimi korvataan muuttujalla

Saadaksemme aikaan tulostuksen kuvaruudulle olemme käyttäneet käskyä PRINT, jota on seurannut toivottu teksti lainausmerkeissä, esim. PRINT ''00120 HELSINKI 12''.

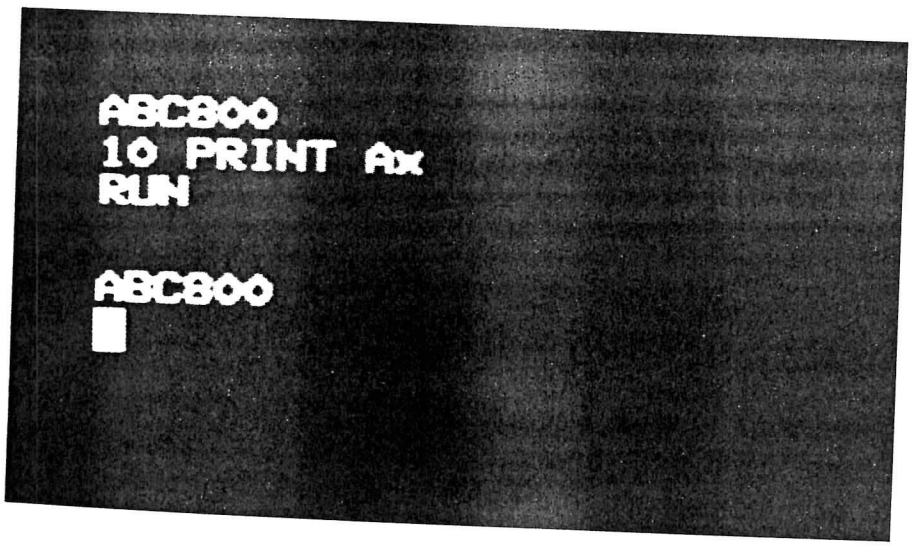
Lainausmerkeissä annettava teksti voidaan korvata yhdellä merkillä. Tällaista merkkiä kutsutaan muuttujaksi. Muuttujaa, joka voi sisältää mitä tahansa tekstiä, kutsutaan merkkijonomuuttujaksi. Yksinkertaisimmassa tapauksessa merkitään merkkijonomuuttujaa yhdellä kirjaimella ja merkillä -☼- ('aurinko'), esim. A-☼-.

Syötä seuraava ohjelma:

10 PRINT A-☼- RETURN

Kuten tiedät, täytyy jokainen syöttö lopettaa painalluksella RETURN . Jatkossa jätämme kuitenkin pois tästä muistuttavan merkin.

Ohjelman ajo komennolla RUN antaa, 'kuten kuvasta näkyy', vain yhden tyhjän rivin.



Tietokone kirjoittaa tyhjän rivin sen vuoksi, että ohjelmassa ei ole vielä annettu merkkijonomuuttujalle A-☼- mitään arvoa.

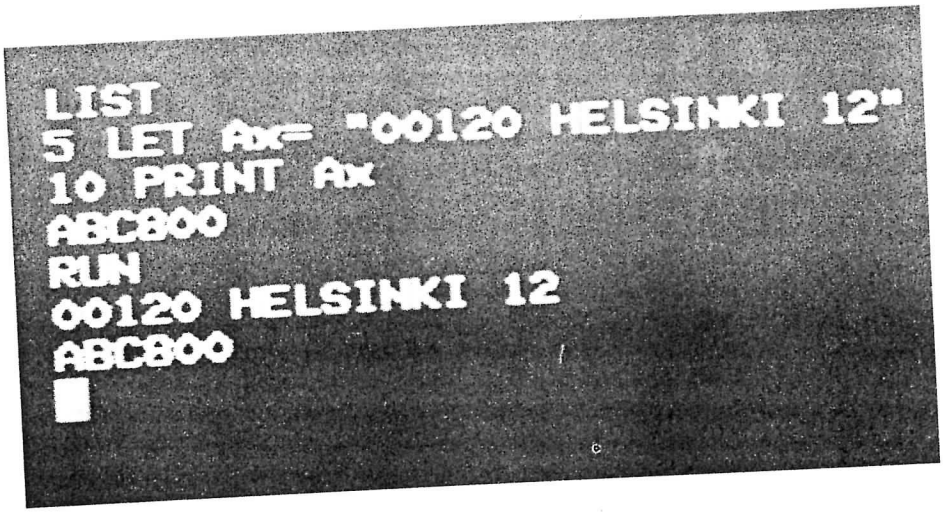
Merkkijonomuuttuja saa arvon

Muuttujalle voidaan antaa arvo useilla eri tavoilla. Aloitamme käyttämällä käskyä LET. Aja sen vuoksi ohjelma, jossa on rivi

```
5 LET A$ = "00120 HELSINKI 12"
```

Merkkijonomuuttujalle A$ annetaan arvo 00120 HELSINKI 12, kun tämä rivi suoritetaan. Ohjelma-ajon aikana pitää muuttuja tämän arvon siihen saakka, kunnes sille annetaan uusi arvo.

Tyhjennä kuvaruutu tavalliseen tapaan käyttämällä näppäimiä CTRL ja L . Muodosta sitten ohjelmalistaus komennolla LIST ja aja sen jälkeen ohjelma.



Kuva osoittaa, että muuttujalle A$ on annettu se arvo, joka tu lostuu kuvaruudulle, kun ohjelmarivi 10 suoritetaan.

Muuttujalle arvo näppäimistön kautta

Useimmissa ohjelmissa vaihtelee muuttujan arvo ajosta toisee Kun me junamatkustajina haluamme paikkalipun, täytyy lipu myyjän näppäinpöytänsä avulla antaa tietokoneelle matkaa k kevat tiedot. Lipun kirjoittamisen hoitava ohjelma VR:n tietol neessa ottaa tietenkin huomioon annetun määränpään, mat ajan jne. Matkaa koskevat tiedot ovat muuttujien arvoja, jo tietokone tarvitsee voidakseen kirjoittaa lipun.

Voidaksemme vastaavalla tavalla antaa muuttujalle A-ö- arvoja tietokoneen näppäimistön kautta, käytämme käskyä INPUT ... Vaihda aikaisempi ohjelmarivi 5 uuteen syöttämällä alla oleva ohjelmarivi samalla rivinumerolla.

```
5 INPUT A-ö-
```

Jotta muuttujien arvojen syöttö ajon aikana helpottuisi, tulee aina esittää kuvaruudulla selventävä teksti. Tällaista tekstiriviä INPUT-lauseen yhteydessä voidaan kutsua alkutekstiksi. Kirjoita rivi:

```
2 PRINT "KIRJOITA NIMI "
```

Täydellinen ohjelma näyttää nyt tällaiselta:

```
2 PRINT "KIRJOITA NIMI "
5 INPUT A-ö-
10 PRINT A-ö-
```

Yhden ajon jälkeen on kuvaruudulla alempana näkyvä teksti. Se teksti, jonka Sinä itse syötät näppäimistöltä, on tässä tavalliseen tapaan merkitty sinisellä.

```
RUN
KIRJOITA NIMI
? MATTI SUOMALAINEN
MATTI SUOMALAINEN
ABC 800
```

Yksinkertainen tapa kirjoittaa alkuteksti

Koska alkuteksti on aina kirjoitettava ennen syöttöä, on olemassa mahdollisuus antaa haluttu alkuteksti suoraan INPUT-lauseessa.

Muuta edellistä ohjelmaa poistamalla rivi 2 ja muuttamalla ohjelmariviä 5 seuraavaan tapaan:

```
2
5 INPUT "KIRJOITA NIMI: "A-ö-
10 PRINT A-ö-
```

Aja ohjelma ja totea, että kysymysmerkkiä ei enää ilmaannu näkyviin ja että rivi ei siirry alkutekstin jälkeen eteenpäin.

Ohjelman alkuun siirtyminen

Jos haluat suorittaa useita samantyyppisiä syöttöjä, voit tietenkin lisätä ohjelmaan monta samanlaista ohjelmakappaletta.

Parempi ratkaisu on antaa tietokoneen suorittaa sama ohjelmakappale useaan kertaan. Tähän päästään mm. käskyllä GOTO ... Kun tietokone suorittaa tämän käskyn, jatkuu ohjelma annetulta ohjelmariviltä.

Täydennä edellistä ohjelmaa lisäämällä ohjelmarivi 20.

```
5 INPUT "KIRJOITA NIMI: "A$
10 PRINT A$
20 GOTO 5
```

Aja ohjelma!

Ohjelma-ajon keskeyttäminen

Kuten ymmärrät, jatkuu tämä ohjelma, kunnes tietokoneen virta katkaistaan tai kunnes painetaan tietokoneen 'hätäkatkaisijaa'. Hätäkatkaisu tapahtuu pitämällä näppäintä CTRL alaspainettuna ja painamalla samanaikaisesti näppäintä C . Tällaista ohjelmanosaa kutsutaan ikuiseksi silmukaksi (loop).

Tietokone antaa katkaisun yhteydessä ilmoituksen, josta selviää millä ohjelmarivillä katkos tapahtui.

```
Stop in Line 5.
```

Jotta ohjelma katkeaisi syötön aikana, tarvitaan vain yksi nappuloiden CTRL ja C painallus. Tämä on erikoistapaus. Tavallisesti tarvitaan ohjelman katkaisemiseksi kaksinkertainen painallus nappuloihin CTRL ja C (katso yhteenvetoa).

Ehdoton hyppy

Ohjelmallamme on se ominaisuus, että ohjelman ajo aina jatkuu riviltä 5 sen jälkeen, kun rivi 20 GOTO 5 on suoritettu. Ohjelmarivit suoritetaan siis järjestyksessä 5, 10, 20, 5, 10, 20 jne. Kun ohjelmarivejä ei suoriteta nousevassa numerojärjestyksessä, on tapana puhua 'hypystä'. Käskyllä GOTO ... suoritetaan aina hyppy annetulle ohjelmariville. Tällaista hyppyä kutsutaan ehdottomaksi hypyksi.

Valinta kahdesta vaihtoehdosta

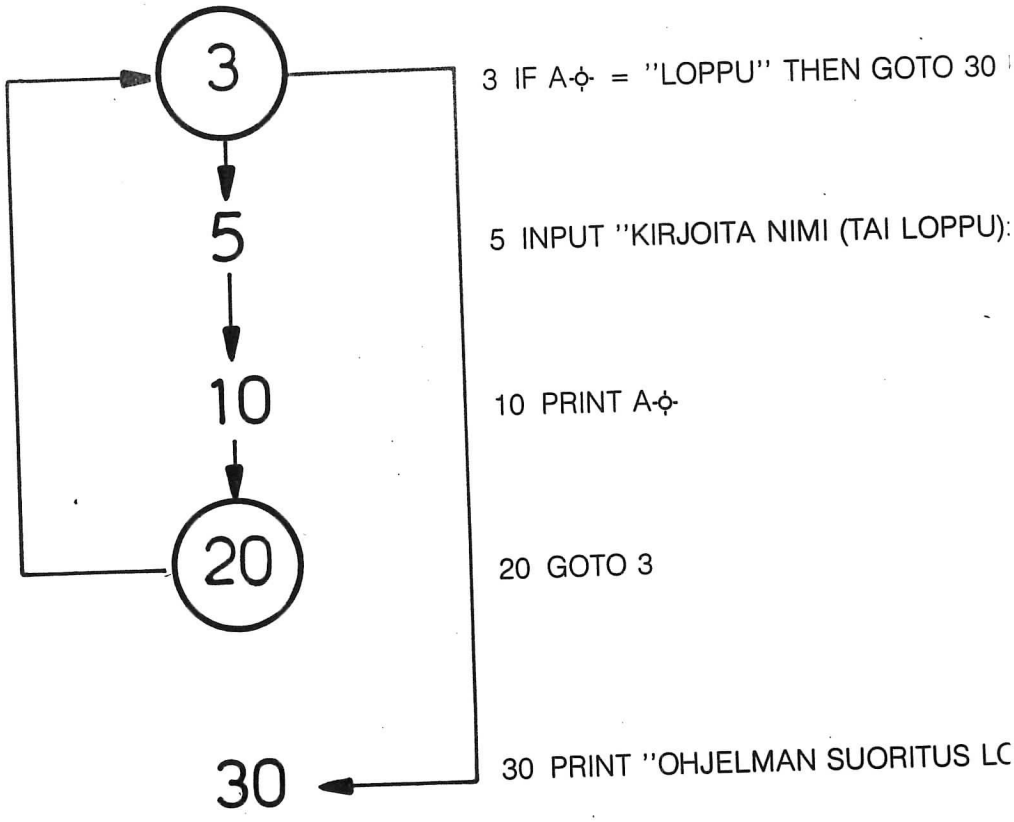
On välttämätöntä, että ohjelma laaditaan niin, että 'hätäkatkaisijaa' ei tarvita. Esimerkissämme tämä voidaan tehdä siten, että ohjelma täydennetään ehdollisella hypyllä. Uusi ohjelmarivi 3 sisältää sen, että hyppy voi tapahtua kahdelle eri ohjelmariville. Rivin alussa oleva ehto määrää mille riville hypätään. Yhtenä vaihtoehtona on syötön toistaminen ja toisena uuden ohjelmarivin 30 suorittaminen.

Ohjelmaesimerkissä on lisäksi alkuteksti ohjelmarivillä 5 sovitettu annetun ehdon mukaiseksi. Koska ehdon täytyy liittyä silmukkaan, on myös ohjelmarivi 20 muutettu.

```
3 IF A-ö- = "LOPPU" THEN GOTO 30 ELSE GOTO 5
5 INPUT "KIRJOITA NIMI (TAI LOPPU):"A-ö-
10 PRINT A-ö-
20 GOTO 3
30 PRINT "OHJELMAN SUORITUS LOPPUNUT"
```

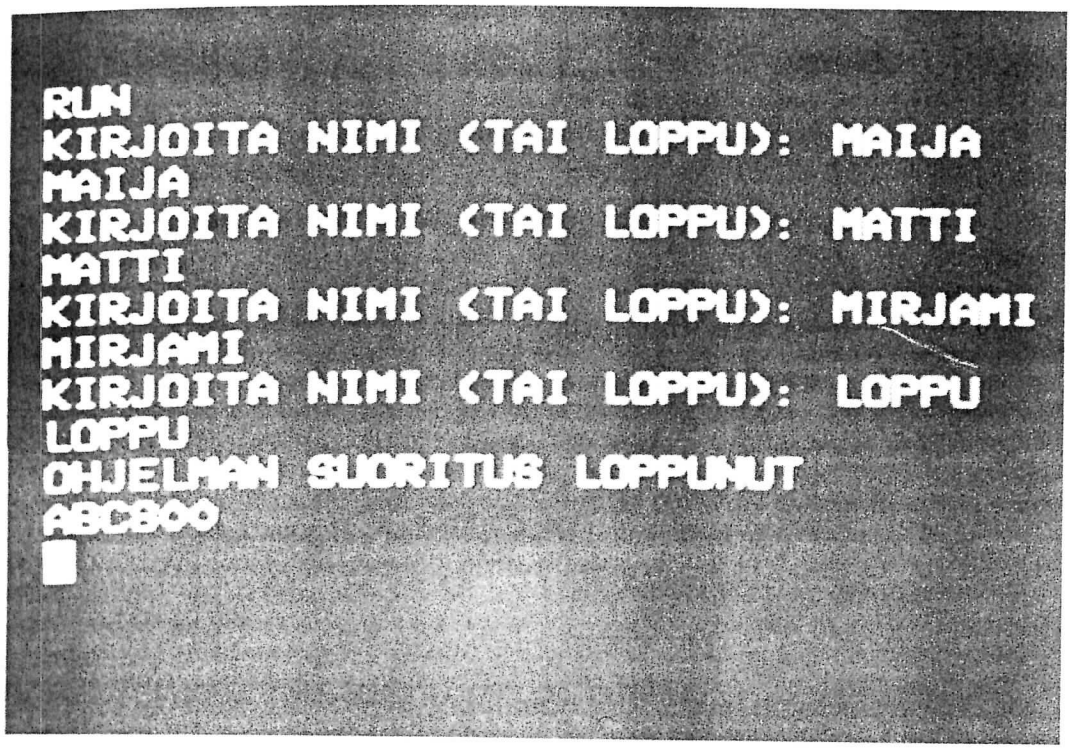
Kun tietokone suorittaa ohjelmarivin 3, se tutkii onko merkkijonomuuttujalle A-ö- annettu arvo 'LOPPU' tai jokin muu arvo, ts. onko A-ö- eri kuin 'LOPPU'.

Suoritetut hypyt käyvät ilmi kuviosta.



Ohjelmarivillä 3 voidaan ELSE GOTO 5 jättää pois, koska ohjelma tämän mukaan jatkuu seuraavalta ohjelmariviltä.

Ohjelman ajo voi antaa esim. seuraavan tuloksen:

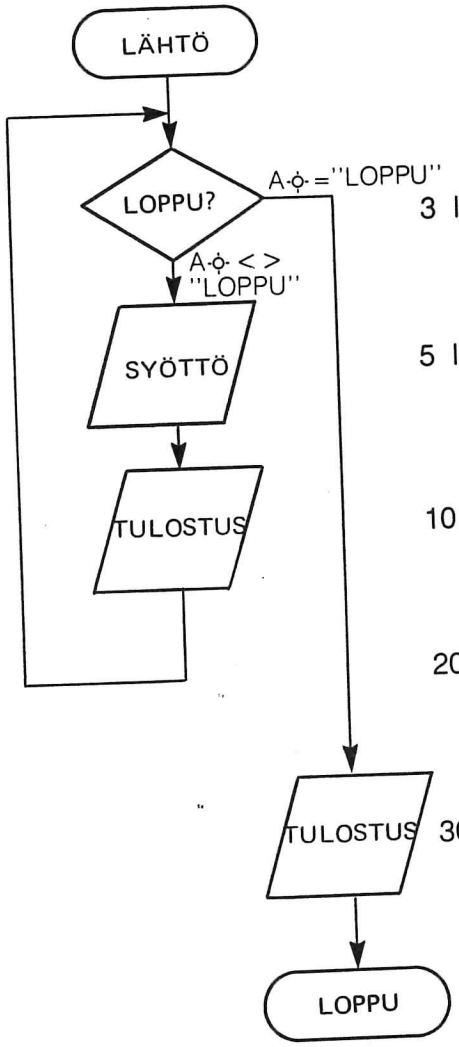


Kun merkkijonomuuttujalle A$ annettiin arvo 'LOPPU', loppui ohjelman suoritus.

Ohjelman kulku vuokaaviolla esitettynä

Ohjelman suunnittelussa käytetään usein ns. vuokaaviota, jossa tapahtumien kulku selviää yhden kuvion avulla. Vertaa vuokaavion symboleja vastaaviin ohjelmariveihin.

Vuokaavio pääsee parhaiten oikeuksiinsa, kun ohjelma on laaja ja kun se sisältää useita ehdollisia ja ehdottomia hyppyjä. Tuonnempana annetaan lisää esimerkkejä vuokaavioiden käytöstä.



3 IF A-◊- = "LOPPU" THEN GOTO 30 ELSE G

5 INPUT "KIRJOITA NIMI (TAI LOPPU): "A-◊-

10 PRINT A-◊-

20 GOTO 3

30 PRINT "OHJELMAN SUORITUS LOPPUN

Parempaa ohjelmointitekniikkaa ehdottomilla hypyillä

Useimmissa tietokonetehtävissä on jaksoja, jotka toistetaan mon ta kertaa. Tällainen jakso toistetaan siihen saakka, kun tietty eht on voimassa. Edellinen ohjelmointiesimerkki tarjosi yhden keino tällaisen ohjelmajakson suorittamiseksi. Seuraavaksi esitettäväll menetelmällä saamme saman tuloksen ja selväpiirteisemmä ohjelman.

Pyyhi tietokoneen muisti komennolla NEW ja syötä sisään seura va ohjelmaesimerkki.

```
10 WHILE A-ö- < > "LOPPU"
20  INPUT "KIRJOITA NIMI (TAI LOPPU):"A-ö-
30  PRINT A-ö-
40 WEND
50 PRINT "OHJELMAN SUORITUS LOPPUNUT"
```

Kun ajat ohjelman huomaat, että ohjelma toimii samoin kuin edellinen esimerkki. Voit myös todeta, että useaan kertaan toistettava ohjelman jakso on pysynyt muuttumattomana.

Uutta on tapa antaa ohjelmajakson laajuus sekä tämän kappaleen toistoehto.

Toistettava ohjelmajakso merkitään alussa käskyllä WHILE ... ja lopussa käskyllä WEND, esimerkissä sisäänvedetyillä ohjelmariveillä 20 ja 30.

Käskyn WHILE perään kirjoitetaan ehto. Ohjelmajaksoa toistetaan niin kauan kuin ehto on tosi, esimerkissämme niin kauan kuin A-ö-:n arvo ei ole "LOPPU". Jotta tietokone osaisi oikein tulkita näitä ehtoja, on käytettävä kirjoitustapaa A-ö- < > "LOPPU".

Kun ohjelmarivin 10 ehto ei enää ole voimassa, jatkuu ohjelman suoritus riville 50.

Useita yhteenkuuluvia muuttujia

Tähän mennessä olemme käyttäneet vain yhtä muuttujaa. Kun halutaan käyttää useita muuttujia, voidaan käyttää merkintöjä B-ö-, C-ö- ja D-ö-. Myös merkinnät A1-ö-, B1-ö- ja C9-ö- ovat sallittuja (katso ohjelmointikortti, sivu 2).

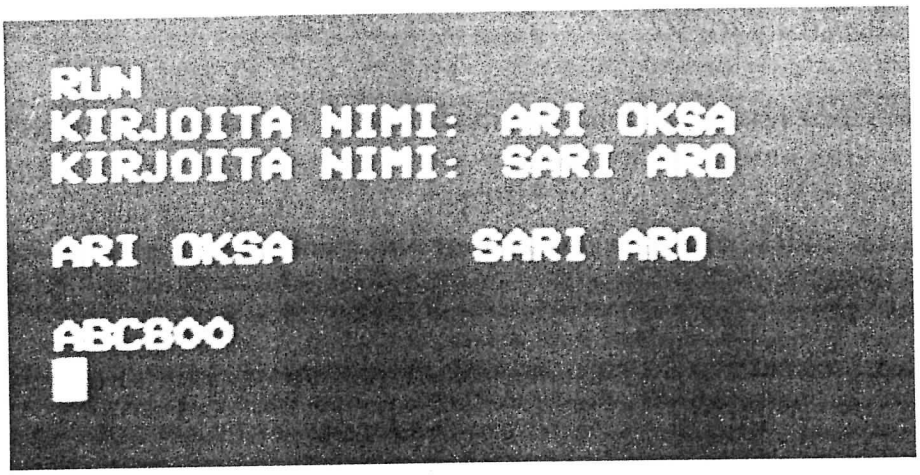
Yhteenkuuluville muuttujille voimme lisäksi käyttää hyväksi mahdollisuutta varustaa muuttuja numerolla, jota kutsutaan indeksiksi, esim. A-ö-(1), A-ö-(2) ja A-ö-(3). Tarkastelkaamme seuraavaa esimerkkiä.

```
10 INPUT "KIRJOITA NIMI: "A$(1)
20 INPUT "KIRJOITA NIMI: "A$(2)
30 PRINT
40 PRINT A$(1),A$(2)
50 PRINT
```

Ohjelmariveillä 10 ja 20 annetaan muuttujille A$(1) ja A$(2) arvot näppäimistöllä tavalliseen tapaan. Kun rivi 40 suoritetaan, kirjoitetaan muuttujien arvot kuvaruudulle. Koska muuttujat erotetaan pilkulla, tapahtuu tulostus sarakkeissa (katso ohjelmointikortin sivulta 7).

Ohjelmarivit 30 ja 50 antavat kumpikin tyhjän rivin. Nämä rivit ovat mukana vain antamassa helposti luettavan tulostuksen.

Ohjelman ajo voi antaa esimerkiksi seuraavan tuloksen.



Käyttäjä päättää muuttujien lukumäärän

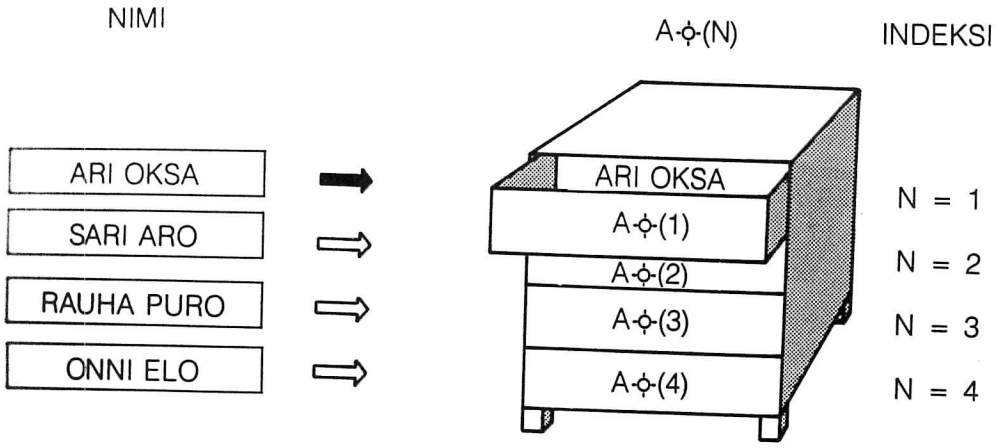
On ilmeisesti suuri etu, jos käyttäjä itse voi päättää kuinka monta muuttujaa on talletettava tietokoneen muistiin. Kuten aikaisemmissa esimerkeissä, voidaan tähän päästä käyttämällä silmukkaa. Syötä seuraavat ohjelmarivit:

```
10 WHILE A$(N)<>"LOPPU"
20  LET N=N+1
30  INPUT "NIMI/(LOPPU): "A$(N)
40 WEND
```

Sitä muuttujaa, jolle annetaan arvoja (nimiä) ohjelmarivillä 30, merkitään A-ö-(N), ja N voi saada arvot 1, 2, 3 jne. N on siis luvun symboli ja sitä kutsutaan lukumuuttujaksi. Niin kauan kuin lukumuuttujalle ei ole annettu muuta arvoa on sillä arvo 0.

Kun silmukkaa suoritetaan ensimmäistä kertaa, annetaan muuttujalle A-ö-(1) arvo (esim. ARI OKSA), ts. lukumuuttujan N arvon tulee olla 1. Toisella kierroksella annetaan muuttujalle A-ö-(2) arvo (SARI ARO) ja lukumuuttuja N saa arvon 2 jne. Lukumuuttujan N kasvattaminen tapahtuu ohjelmarivillä 20, jossa sen arvo kasvaa 1:llä jokaisella silmukan kierroksella.

Seuraava luonnos auttaa ymmärtämään, miten tapahtuu nimien tallentuminen tietokoneen muistiin merkkijonomuuttujan avulla.



Vastaavan ohjelmasilmukan avulla on nyt tarkoituksena saada kuvaruudulle ne nimet, jotka on tallennettu tietokoneen muistiin.

Muuttujan N arvo on asetettava nollaksi, jotta tulostukseen olisi käytettävissä samankaltainen silmukka. Kun ohjelmarivi 50 suoritetaan, saa N arvon nolla.

```
50 LET N = 0
60 WHILE A-ö-(N)< >"LOPPU"
70  LET N= N+1
80  PRINT A-ö-(N)
90 WEND
```

Muuttujan N loppuarvo vaihtelee sen mukaan kuinka monta nimeä on tietokoneen muistiin tallennettu. Jos on tallennettu neljä nimeä, on N ajon jälkeen 4.

Jos halutaan tallentaa enemmän kuin 9 nimeä, on tämä 'selvitettävä' tietokoneelle. Asia järjestyy käyttämällä käskyä DIM ... Palaamme tuonnempana tähän käskyyn.

Aja ohjelmaa muutamia kertoja ja anna syötettyjen nimien lukumäärän vaihdella (kuitenkin korkeintaan 9 kpl). Muista, että nimien syöttö loppuu, kun sana LOPPU on annettu sisään.

Pieni rekisteri

Näyttääksemme kuinka yksinkertainen rekisteri voidaan laatia, tallennamme pienen puhelinluettelon tietokoneen muistiin. Vertaa edelliseen ohjelmaan niitä rivejä, jotka seuraavassa on painettu sinisellä.

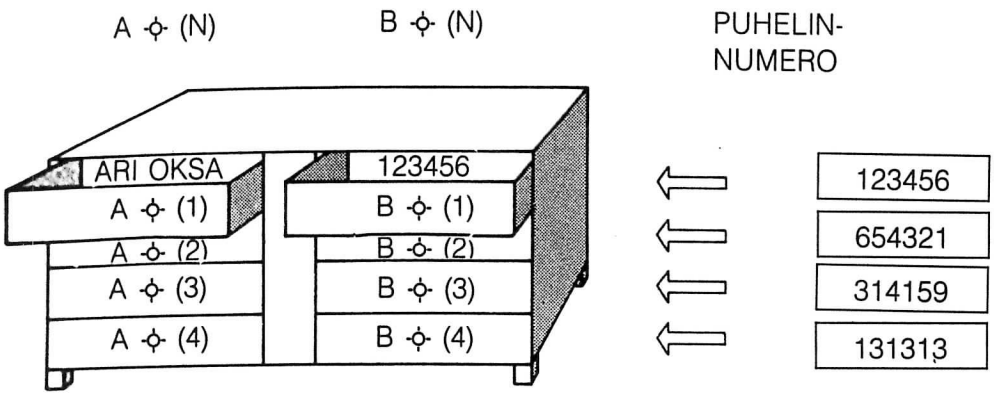
```
10 WHILE A$(N)<>"LOPPU"
20  LET N=N+1
30  INPUT "NIMI/(LOPPU): "A$(N)
35  INPUT "PUHELIN:      "B$(N)
40 WEND

50 LET N=0
60 WHILE A$(N)<>"LOPPU"
70  LET N=N+1
80  PRINT A$(N),B$(N)
90 WEND

100 INPUT "UUSI TULOSTUS? (KYLLÄ/EI) "V$
110 IF V$="KYLLÄ" THEN GOTO 50
120 PRINT "OHJELMAN SUORITUS LOPPUNUT"
```

SYÖTTÖ

TULOSTUS

UUSI TULOSTUS

Puhelinluettelon syöttö tapahtuu ohjelmariveillä 10—40 olevan silmukan avulla. Puhelinnumerot tallennetaan indeksoituun muuttujaan B$(N). Kun N = 1, annetaan muuttujalle A$(1) ensimmäinen nimi ja muuttujalle B$(1) ensimmäinen puhelinnumero jne. Seuraavasta kuviosta ilmenee, kuinka yhteenkuuluvat nimet ja puhelinnumerot tallennetaan.



Vastaavalla tavalla tapahtuu puhelinluettelon kirjoittaminen, kun ohjelmarivit 50—90 suoritetaan. Rivit 100—120 on lisätty, jotta olisi mahdollista kirjoittaa puhelinluettelo useita kertoja.

Kun ajat ohjelman, voi kuvaruutu näyttää seuraavalta.

```
RUN
NIMI/<LOPPU>: ARI OKSA
PUHELIN:      123 456
NIMI/<LOPPU>: SARI ARO
PUHELIN:      654 321
NIMI/<LOPPU>: RAUHA PURO
PUHELIN:      314 159
NIMI/<LOPPU>: ONNI ELO
PUHELIN:      131 131
NIMI/<LOPPU>: LOPPU
PUHELIN:
ARI OKSA      123 456
SARI ARO      654 321
RAUHA PURO    314 159
ONNI ELO      131 131
LOPPU
UUSI TULOSTUS? <KYLLÄ/EI> ■
```

Jos muuttujalle V-ọ- annetaan riviä 100 suoritettaessa arvo ''KYLLÄ'', toteutuu rivin 110 ehto ja puhelinluettelo kirjoitetaan uudelleen. Tämä on mahdollista, koska kaikki muuttujien arvot ovat nyt tallella tietokoneen muistissa.

Jos muuttujalla V-ọ- on muu arvo kuin ''KYLLÄ'' riviä 110 suoritettaessa, jatkuu ohjelman suoritus riville 120. Ohjelman ajo loppuu tavalliseen tapaan viimeisen rivin tultua suoritetuksi.

Kun ohjelma ajetaan uudelleen, täytyy kaikki nimet ja puhelinnumerot syöttää vielä kerran. Tätä voidaan pitää haittana. Tuonnempana kirjassa palataan tämän ongelman ratkaisemiseen.

Haku rekisteristä

Rekisteristä tulee käyttökelpoinen vasta sitten, kun sieltä voi helposti etsiä haluttua tietoa. Meidän rekisteriämme voi käyttää tietyn henkilön puhelinnumeron etsimiseen. Kun ohjelman aikana annetaan nimi, tulee tietokoneen esittää vastaava puhelinnumero.

Nimen ja puhelinnumeron syöttö tapahtuu esimerkissämme samanaikaisesti ja näillä molemmilla muuttujilla on sama indeksi (muuttujan N arvo). Kun tietokone löytää oikean nimen, A-ǫ-(N), on rekisterissä saatavana vastaava puhelinnumero muuttujan B-ǫ-(N) arvona samalla indeksin N arvolla.

Etsintä tapahtuu siten, että tietokone vertailee toinen toisensa perään annettua nimeä niihin nimiin, jotka ovat rekisterissä. Jos asianomainen nimi on rekisterissä, tulostetaan nimi ja etsitty puhelinnumero kuvaruudulle.

Ne harvat täydennykset, jotka edelliseen ohjelmaan on täytynyt suorittaa, on merkitty sinisellä seuraavaan ohjelmalistaukseen.

```
10 WHILE A-ǫ-(N)< >"LOPPU"                          } SYÖTTÖ
20  LET N=N+1
30  INPUT "NIMI/(LOPPU): "A-ǫ-(N)
35  INPUT "PUHELIN:      "B-ǫ-(N)
40 WEND

50 LET N=0                                          } HAKU
55 INPUT "KENEN PUHELINNUMERON HALUAT? "V-ǫ-
60 WHILE A-ǫ-(N)< >"LOPPU"
70  LET N=N+1 : REM"
80  IF V-ǫ-=A-ǫ-(N) THEN PRINT A-ǫ-(N),B-ǫ-(N)
90 WEND

100 INPUT "UUSI HAKU? (KYLLÄ/EI) "V-ǫ-             } UUSI
110 IF V-ǫ-="KYLLÄ" THEN GOTO 50                    } HAKU
120 PRINT "OHJELMAN SUORITUS LOPPUNUT"
```

Kun ohjelmarivi 55 on suoritettu, on muuttuja V-ϕ- = se nimi jota haku koskee. Aina kun rivi 80 suoritetaan, vertaillaan muuttujaa V-ϕ- yhteen rekisterin nimeen. Etsintä jatkuu koko rekisterin läpi. Kun tietokone löytää oikean nimen rekisteristä, kirjoitetaan nimi, A-ϕ-(N), ja vastaava puhelinnumero, B-ϕ-(N), kuvaruudulle. Puhelinluettelon syötön ja kahden haun jälkeen voi kuvaruutu näyttää seuraavalta.

```
RUN
NIMI/(LOPPU): ARI OKSA
PUHELIN:      123 456
NIMI/(LOPPU): SARI ARO
PUHELIN:      654 321
NIMI/(LOPPU): RAUHA PURO
PUHELIN:      314 159
NIMI/(LOPPU): ONNI ELO
PUHELIN:      131 131
NIMI/(LOPPU): LOPPU
PUHELIN:
KENEN PUHELINNUMERON HALUAT? SARI ARO
SARI ARO      654 321
UUSI HAKU? (KYLLÄ/EI) KYLLÄ
KENEN PUHELINNUMERON HALUAT? ONNI ELO
ONNI ELO      131 131
UUSI HAKU? (KYLLÄ/EI) EI
OHJELMAN SUORITUS LOPPUNUT
ABC800
```

Näppäimistö

CE Poistaa kirjoitetun rivin.

CTRL L Siirtää kohdistimen vasempaan yläkulmaan ja tyhjentää kuvaruudun.

CTRL C Keskeyttää meneillään olevan ohjelman ajon. Tätä voi seurata jokin seuraavista:

- ▢ Ohjelman ajo jatkuu (valinnainen näppäin).
- CTRL S Kullakin näppäimen painalluksella suoritetaan yksi ohjelmarivi.
- CTRL C Ohjelman ajo lopetetaan.

Funktionäppäimen toiminta on kuvattu käyttöohjeessa.

Käskyjä

GOTO ...

Aiheuttaa hypyn ilmoitetulle riville.

```
60 GOTO 10
```

IF ... THEN ... ELSE ...

Jos IF-sanan jälkeinen ilmaus on tosi (≠ 0), silloin suoritetaan se, mitä THEN- ja ELSE-sanojen väliin on kirjoitettu. Muussa tapauksessa suoritetaan se, mitä on kirjoitettu ELSE-sanan jälkeen. ELSE ... voidaan jättää pois.

```
40 IF A = 0 THEN GOTO 60 ELSE 10
```

(GOTO voidaan jättää pois THEN- ja ELSE-sanojen välissä.)

INPUT "..." ...

Kirjoittaa annetun alkutekstin kuvaruutuun ja odottaa sen jälkeen tiedon syöttöä näppäimistön kautta. Alkuteksti voidaan jättää pois.

```
30 INPUT "ANNA NIMESI "V$
90 INPUT P
```

LET ...

Sijoittaa arvon muuttujalle. LET voidaan jättää pois.

```
10 LET X = 5 * 5
20 T$ = "LEENA"
```

WHILE ...

WEND

Ohjelmajaksoa toistetaan niin kauan kuin annettu ehto on tosi.

```
10 WHILE X < 10
15   LET X = X + 1
20   PRINT X
30 WEND
```

Harjoitustehtäviä

* 3.1 Täydennä seuraavaa ohjelmaa

```
10 LET A-ö- = "OSAAN"
20 LET B-ö- = "MINÄ"
30 LET C-ö- = "OHJELMOIDA"
```

niin, että kuvaruutuun tulee seuraava tulostus ajon jälkeen.

* 3.2 Mikä muutos on tehtävä seuraavaan ohjelmaan, jotta tulostus tulisi sarakkeisiin?

```
10 INPUT "KIRJOITA KAUPUNGIN NIMI: "A-ö-
20 WHILE N<60
30  N=N+1
40  PRINT A-ö-;
50 WEND
```

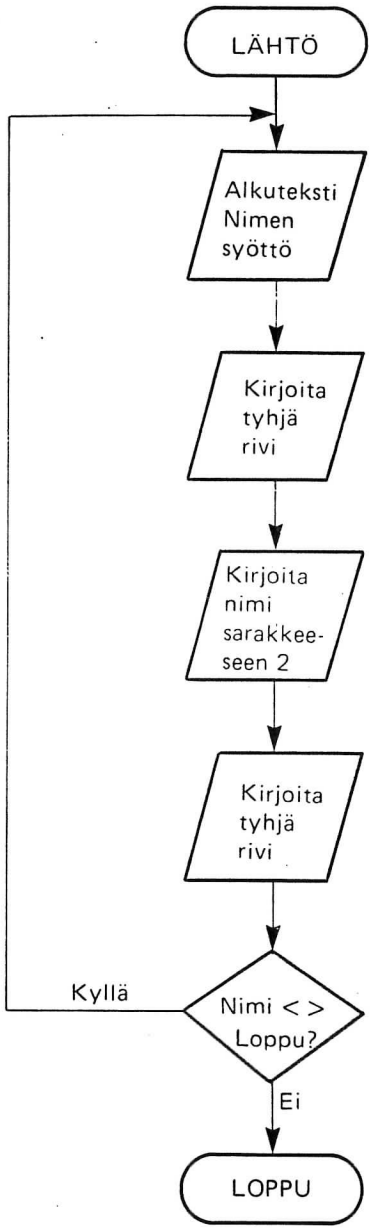
* 3.3 Kun ohjelmarivi 10 suoritetaan, annetaan näppäimistön avulla muuttujalle A-ö- arvo, esim. Rovaniemi. Täydennä ohjelmaa niin, että kuvaruudulle tulostuu kaupungin nimen perään teksti "ON KAUNIS KAUPUNKI".

```
10 INPUT "KIRJOITA KAUPUNGIN NIMI: "A-ö-
```

* 3.4 Kun ajat seuraavaa ohjelmaa ja vastaat oikein kysymykseen, tulostuu teksti "OIKEA KAUPUNKI". Muuta yhtä ohjelmariviä niin, että saat tulostuksen "VÄÄRÄ MAA", jos vastaat väärin asetettuun kysymykseen.

```
10 PRINT "MINKÄ NIMINEN ON NORJAN"
20 PRINT "PÄÄKAUPUNKI? "S-ö-

30 PRINT
40 IF S-ö-="OSLO" THEN PRINT "OIKEA KAUPUNKI"
50 PRINT
```

* 3.5 Laadi ohjelma, joka noudattelee seuraavaa vuokaaviota.



* 3.6 Erään yrityksen byrokratisoitumisprosessin helpottamiseksi pyydettiin ohjelmaa, joka 'piristäisi' työtovereiden nimittelykäytäntöä. Tähän päästään liittämällä jokaisen 'tittelin' eteen liite BYRO.

Kirjoita ohjelma, joka suorittaa tämän automaattisesti. Testaa ohjelmaa esim. seuraavilla syötöillä: JOHTAJA, SIHTEERI, LÄHETTI.

* 3.7 Parantaakseen asiakaspiirin mahdollisuuksia tavoittaa virkailijoita hankkii Kongressipalvelu Oy tietokoneen, joka asetetaan puhelinvaihteeseen. Tarkoituksena on tallentaa tietokoneeseen tiedot siitä, missä virkailijat kulloinkin ovat.

Aja seuraava ohjelma ja syötä 8 virkailijan nimet. Tämän jälkeen voidaan virkailijoiden 'asema' aina lukea ja muuttaa.

```
100 PRINT
110 PRINT "LÄSNÄOLOLISTA — SYÖTTÖ"
120 WHILE A$(N)<>"LOPPU"
130  LET N=N+1
140  INPUT "NIMI (LOPPU):"A$(N)
150  LET B$(N)="TILAPÄISESTI LÄSNÄ"
160 WEND
170 PRINT
180 INPUT "KENEN SIJAINTIA TOIVOTAAN ?: "S$
190 LET N=0
200 WHILE A$(N)<>"LOPPU"
210  LET N=N+1
220  IF S$<>A$(N) THEN 280
230  PRINT
240  PRINT A$(N),B$(N)
250  INPUT "UUSI SIJAINTI? (KYLLÄ/EI) ";T$
260  IF T$<>"KYLLÄ" THEN GOTO 280
270  INPUT "ANNA UUSI SIJAINTI: "B$(N)
280 WEND
290 PRINT
300 PRINT
310 INPUT "UUSI HAKU? (KYLLÄ/EI) ";S$
320 IF S$<>"EI" THEN GOTO 170
```

Jotta ohjelma tulisi käyttökelpoiseksi suuremmalle määrälle virkailijoita, täytyy lisätä seuraavat ohjelmarivit:

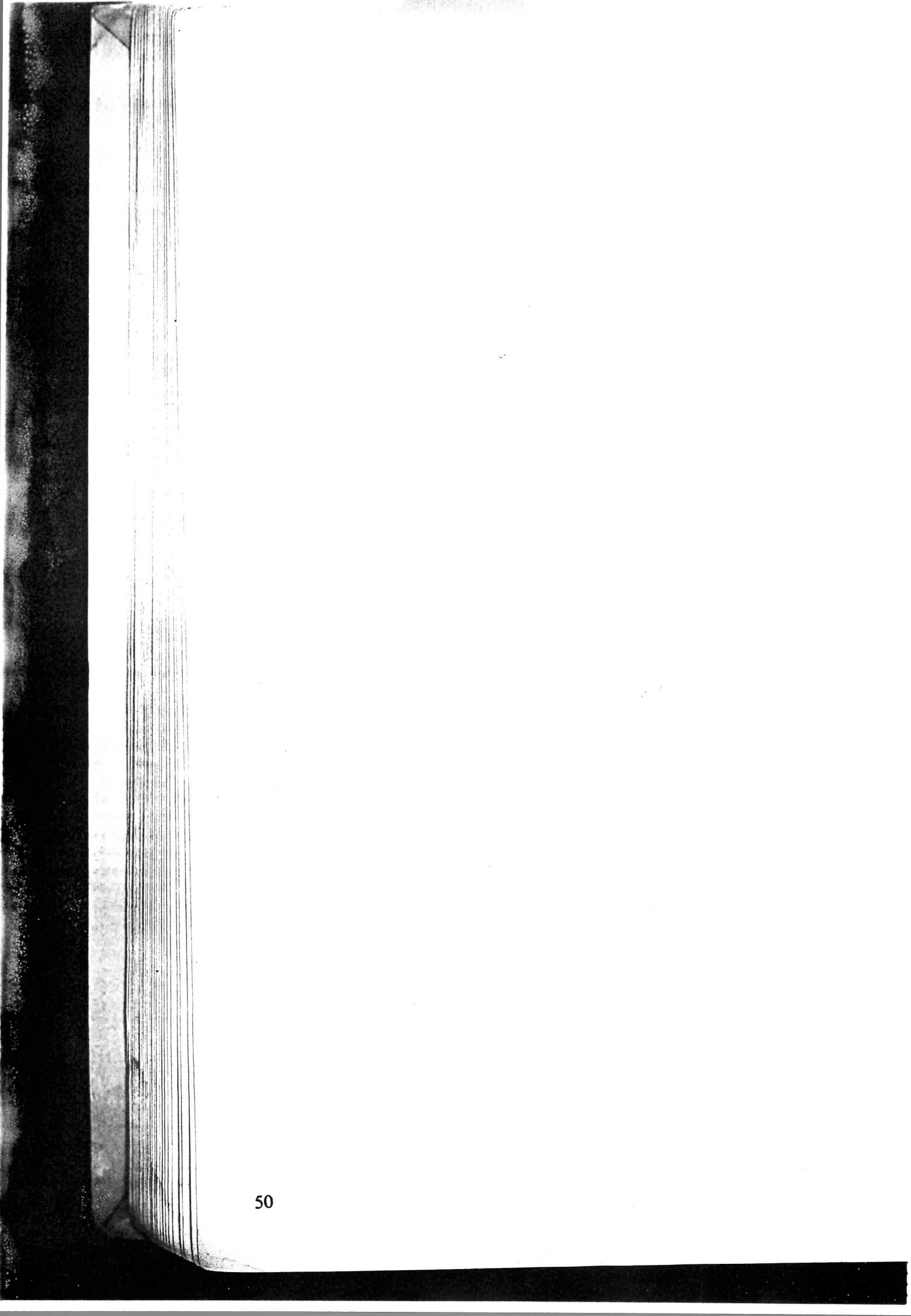
```
80 DIM A$(200)=30
90 DIM B$(200)=30
```

Näiden avulla varataan tilaa tietokoneen muistista 200 henkilölle ja tiedolle: nimellä ja sijainnilla saa kummallakin olla korkeintaan 30 merkkiä.

Käy läpi ohjelma ja yritä selvittää kuinka se toimii.

4

Tietokone laskee



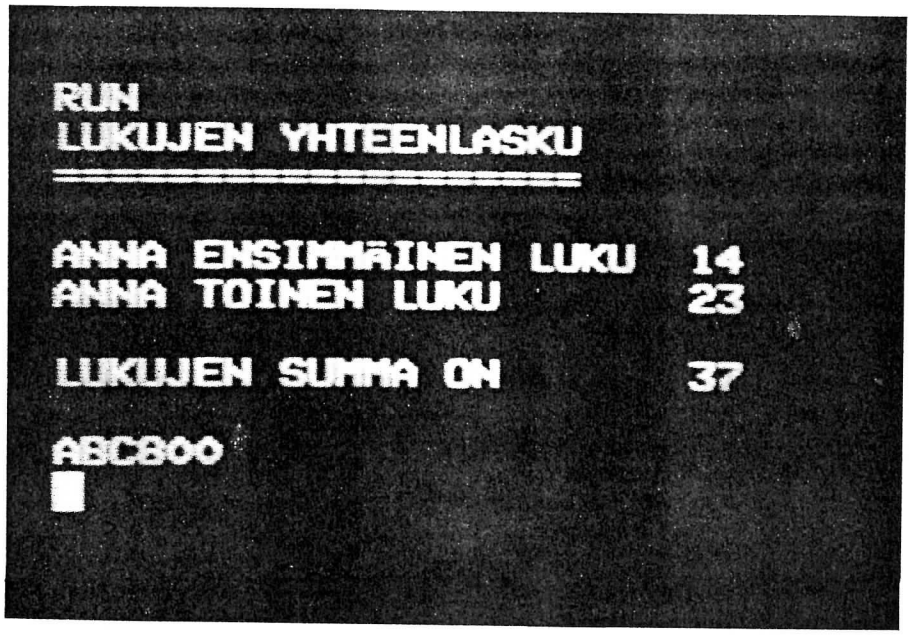
Lukumuuttujia laskuja varten

Tavallisia matemaattisia laskuja, esim. yhteen- ja vähennyslaskuja, voidaan suorittaa vain käyttämällä lukuja. Luku voidaan kirjoittaa joko lukujen tai muuttujien avulla. Muuttujaa, joka voi merkitä vain lukua, kutsutaan lukumuuttujaksi ja sitä merkitään yksinkertaisimmassa tapauksessa yhdellä kirjaimella.

Ennen seuraavaa harjoitusta on Sinun pyyhittävä tietokoneen muisti tavalliseen tapaan komennolla NEW. Seuraava ohjelma kuvaa lukumuuttujien käyttöä. Syötä ohjelma tietokoneeseen ja aja se.

```
10 PRINT "LUKUJEN YHTEENLASKU"
20 PRINT "=============="
30 PRINT
40 INPUT "ANNA ENSIMMÄINEN LUKU   "A
50 INPUT "ANNA TOINEN LUKU        "B
60 PRINT
70 PRINT "LUKUJEN SUMMA ON        " A+B
80 PRINT
```

Laskelma tulostuu seuraavalla tavalla.



Kotkan
kaupunginkirjasto

Kuvaruudun tyhjentäminen ajon aikana

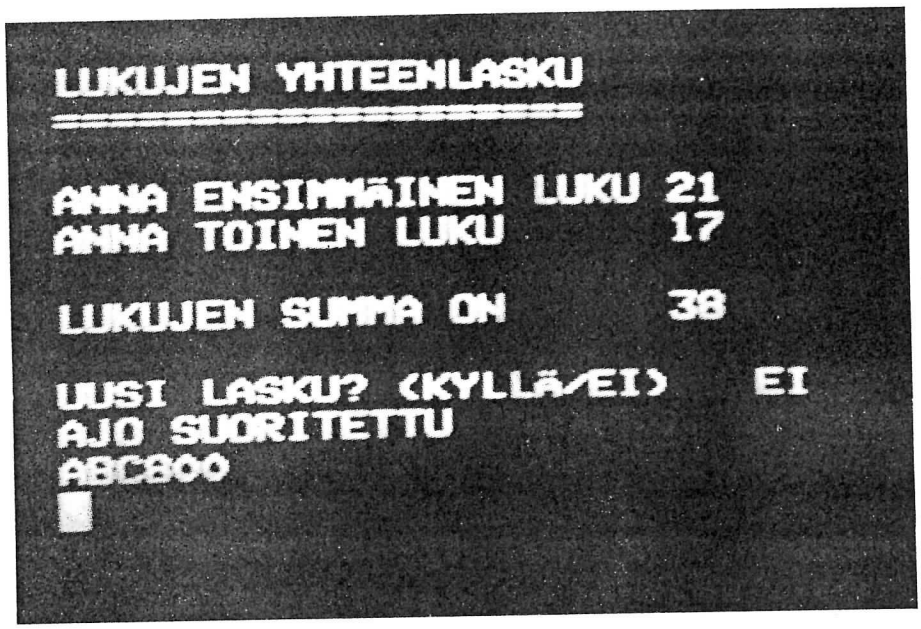
On usein suotavaa, että tietokone esittää tietoa tiettyyn kuvaruudun kohtaan. Tähän päästään esim. tyhjentämällä kuvaruutu. Kuvaruutu voidaan tyhjentää ohjelma-ajon aikana käyttämällä lausetta PRINT CHR-ö-(12).

Muuta edellistä ohjelmaa lisäämällä sinisellä painetut rivit. Aja ohjelma ja anna tietokoneen suorittaa useita yhteenlaskuja.

```
5 PRINT CHR-ö-(12)
10 PRINT "LUKUJEN YHTEENLASKU"
20 PRINT "= = = = = = = = = = = = = = = ="
30 PRINT
40 INPUT "ANNA ENSIMMÄINEN LUKU  "A
50 INPUT "ANNA TOINEN LUKU       "B
60 PRINT
70 PRINT "LUKUJEN SUMMA ON       " A+B
80 PRINT
90 INPUT "UUSI LASKU? (KYLLÄ/EI)  "S-ö-
100 IF S-ö-<>"EI" THEN GOTO 5
110 PRINT "AJO SUORITETTU"
```

Huomaa, että kuvaruudun tyhjennyksen jälkeen alkaa tulostus kuvaruudun ylälaidasta.

Niin kauan kuin S-ö- rivillä 90 ei saa arvoa "EI" on rivin 100 ehto tosi ja ohjelman suoritus jatkuu riville 5. Ohjelman suoritus loppuu, kun S-ö- saa arvon "EI".

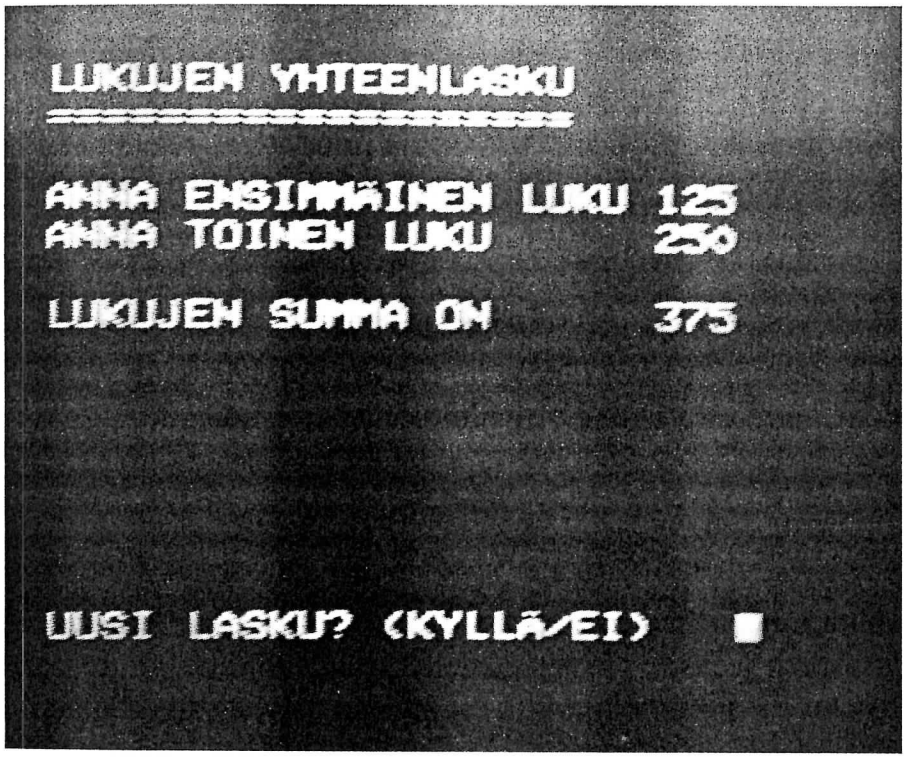


Tulostus haluttuun kohtaan

Kuvaruudun käytön suurena etuna on se, että tieto voidaan esittää hyvin nopeasti. Myös kuvaruudun tyhjennys sujuu silmänräpäyksessä. Toisena etuna on se, että tulostus voi tapahtua 24 riville mielivaltaisessa järjestyksessä. Tähän päästään antamalla toivottu rivi PRINT-lauseessa. Samalla annetaan riviltä myös kohta, josta tulostuksen tulee alkaa.

On usein tarkoituksenmukaista antaa käyttäjälle suoritusohjeita ja valintakysymyksiä tietyssä kuvaruudun kohdassa. Kokeile esim. edellistä ohjelmaa muutettuasi ohjelmariviä 80 seuraavasti:

```
80 PRINT CUR(18, 0);
```



rivi 18→

Lisäys CUR(18,0) PRINT-lauseeseen rivillä 80 aiheuttaa kohdistimen (engl. CURsor) sijoittumisen riville 18 sarakkeeseen 0. Erikoisfunktion CUR(R,K) avulla voidaan kohdistin sijoittaa toivottuun paikkaan.

Sarakkeiden lukumäärä riippuu käytössä olevasta kuvaruudusta.

PRINT CUR(R,K)

Rivi R
Rivit 0—23
24 riviä sekä
ABC800C:lle että ABC800M:lle

Sarake K
Sarakkeet 0—39/79
40 saraketta ABC800C:lle
80 saraketta ABC800M:lle

Kun PRINT-lause lopetetaan puolipisteellä poistuu rivinsiirto, joka muussa tapauksessa liittyy PRINT-lauseen suoritukseen.

Tästä seuraa että alkuteksti, joka tulostuu kun ohjelmarivi 90 suoritetaan, sijoittuu kuvaruudulle riville 18 ja sarakkeeseen 0.

Tietokoneen editointinappulat

Ohjelmointikieli BASIC koostuu joukosta sääntöjä tietokonee ohjelmoimiseksi. Ellei näitä sääntöjä noudateta, reagoi tietokon virheilmoituksella. Yritä lisätä edelliseen ohjelmaan seuraava riv

```
120 PRINT "PÖTYÄ"
```

Mitä tapahtuu? Koska tietokone ei pysty tulkitsemaan virheellis käskyä, antaa tietokone virheilmoituksen, esim.

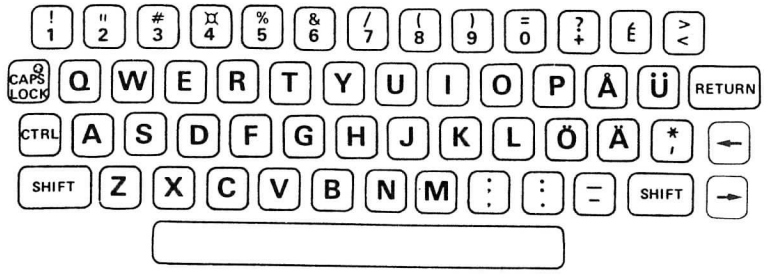
Error 220.

Ohjelmointikortin sivun 10 mukaan virhekoodi 220 tulkita 'Syntaksivirheeksi'. Tiettyjen edellytysten vallitessa antaa tietol ne tämän sijaan virheilmoituksen suorasanaisesti, tässä tapauks sa

Syntaksivirhe

Kun tietokone antaa virheilmoituksen, ei ohjelmariviä hyväks Vaikka tietokone ei hyväksy virheellistä ohjelmariviä, ei muu

tua riviä tarvitse uudelleen syöttää, vaan korjaus voidaan suorittaa käyttämällä tietokoneen editointinäppäimiä. Nämä näppäimet ovat kopiointinäppäin → ja poistonäppäin ←



Editointi sujuu tässä tapauksessa seuraavasti:

120 PRU	→	kopioi 7 merkkiä
120 PR	←	poista 1 merkki
120 PRI	I	kirjoita kirjain I
120 PRINT	→	kopioi muut merkit
	RETURN	lopeta rivi

Mukava tapa korjata ohjelmarivi

Tekniikka, jota käytettiin virheellisten ohjelmarivien korjaamiseen, soveltuu myös oikeiden ohjelmarivien muuttamiseen. Muuttakaamme esimerkiksi aikaisempaa yhteenlaskuohjelmaa siten, että siitä tulee kertolaskuohjelma.

Komennolla ED tulostetaan editoitava (engl. EDit) ohjelmarivi kuvaruudulle. Tämän jälkeen käytetään editointinäppäimiä tavalliseen tapaan.

Tuo rivi 10 ja muuta yhteenlasku kertolaskuksi.

```
ED 10
10 PRINT "KAHDEN LUVUN YHTEENLASKU"
10 PRINT "KAHDEN LUVUN KERTOLASKU"
```

Suorita vastaavaan tapaan merkityt muutokset seuraaville riveille.

```
20 PRINT "= = = = = = = = = = = = = = = = = = = = = = = = = ="
70 PRINT "LUKUJEN TULO ON " A*B
90 INPUT "TAHDOTKO JATKAA? (KYLLÄ/EI) "S-
```

Jos Sinulla on 40 sarakkeen kuvaruutu, ei ohjelmarivi 90 mahdu kuvaruudulle korjauksen jälkeen. Ohjelmarivi 90 saa seuraavan ulkomuodon kuvaruudulla:

```
90 INPUT "TAHDOTKO JATKAA? (KYLLÄ/EI) "S
-
```

Tämä ohjelmarivi on sallittu ABC800C:ssä, sillä ohjelmarivissä voi olla merkkejä aina 160 merkkiin asti. Ohjelmarivi voi koostua siis neljästä kuvaruudun rivistä.

Kaikkien rivinumeroiden muuttaminen

Muutosten ja lisäysten jälkeen on toisinaan välttämätöntä numeroida uudelleen (engl. RENumber) ohjelman rivit. Tämä tapahtuu käyttämällä komentoa REN. Rivien keskinäisen järjestyksen tulee säilyä, mutta ensimmäinen rivinumero ja rivinumeroiden väli on valittavissa. Kokeile antamalla kertolaskuohjelmalle uudet rivinumerot komennolla:

```
REN 200 20
```

Tulos:

```
200 PRINT CHR$(12)
220 PRINT "LUKUJEN KERTOLASKU"
240 PRINT "= = = = = = = = = = = = = ="
260 PRINT
280 INPUT "ANNA ENSIMMÄINEN LUKU "A
300 INPUT "ANNA TOINEN LUKU      "B
320 PRINT
340 PRINT "LUKUJEN TULO ON       " A*B
360 PRINT CUR(18,0);
380 INPUT "HALUATKO JATKAA? (KYLLÄ/EI) "S$
400 IF S$<>"EI" THEN GOTO 200
420 PRINT "AJO SUORITETTU"
```

Useampia muunnelmia REN-komennosta on olemassa. Katso tarkemmin yhteenvedosta.

Tyhjennä nyt tietokoneen muisti komennolla NEW.

Automaattinen rivinumerointi

Ohjelmointityön helpottamiseksi voidaan tietokoneelle antaa tehtäväksi automaattisesti varustaa ohjelmarivit rivinumeroilla. Tähän päästään antamalla komento AUTO, joka sisältää sen, että rivinumeroiden väliksi tulee 10.

```
AUTO
10
```

Tietokone esittää ensimmäisen rivinumeron, joka tässä tapauksessa on 10, ja odottaa sen jälkeen että varsinainen ohjelmarivi syötetään näppäimistön avulla. Kun painat return-näppäintä heti rivinumeron jälkeen, lakkaa automaattinen rivinumerointi.

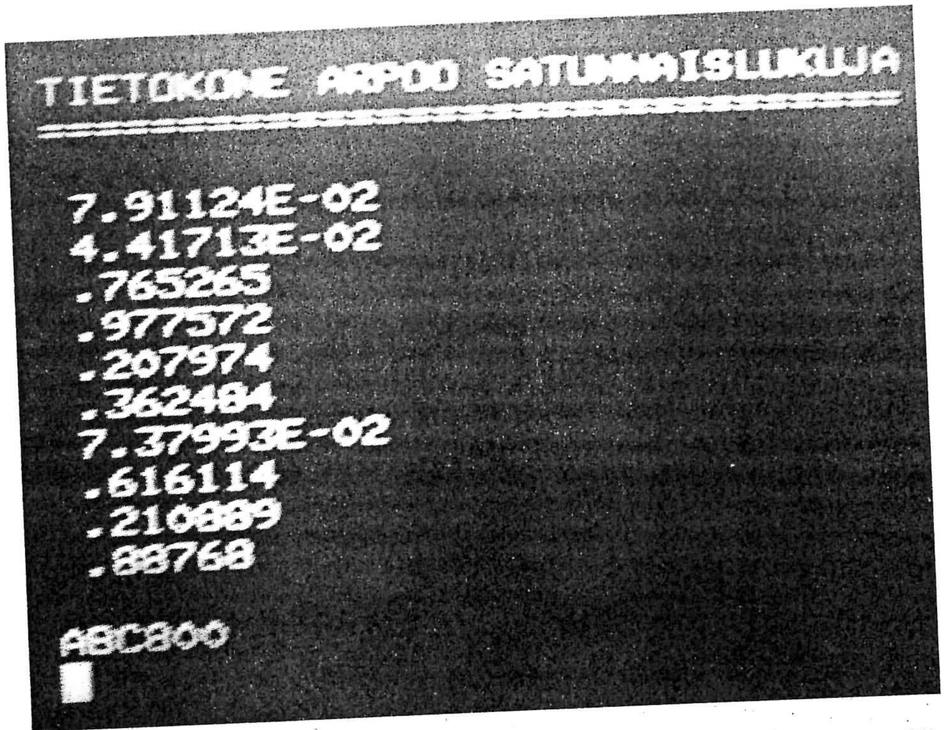
On olemassa useita mahdollisuuksia käyttää komentoa AUTO. Katso tarkemmin yhteenvedosta. Voit tietenkin aina käyttää automaattista rivinumerointia, mutta myös jatkossa merkitsemme värillä ne rivinumerot, jotka tulee syöttää näppäimistöä käyttämällä.

Tietokone arpoo satunnaislukuja

Monissa tilanteissa on tarvetta käyttää umpimähkään valittuja lukuja. Tällaisia satunnaislukuja voidaan tietokoneella muodostaa käyttämällä funktiota RND. Ne satunnaisluvut, jotka muodostetaan ovat lukujen 0 ja 1 välillä. Seuraavan ohjelman avulla voit tuottaa 10 ensimmäistä satunnaislukua.

```
10 PRINT CHR$(12)
20 PRINT "TIETOKONE ARPOO SATUNNAISLUKUJA"
30 PRINT "==================="
40 PRINT
50 WHILE N<10
60  N=N+1
70  PRINT RND
80 WEND
90 PRINT
```

Ajamalla ohjelman useita kertoja voit todeta, että saat aina samat satunnaisluvut. Kuvaruutu näyttää seuraavalta:



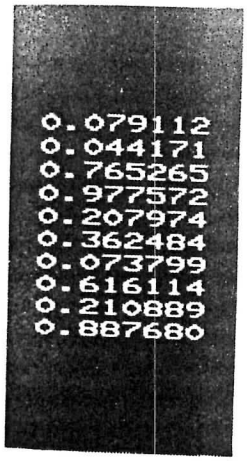
Tietokoneen tapa esittää pieniä lukuja voi vaikuttaa kummalliselta. Ensimmäinen satunnaisluku on 0,0791124 ja tietokone esittää sen muodossa 7.91124E-02. Piste vastaa desimaalipilkkua. Lisäys E-02 sisältää sen, että luku 7,91124 täytyy kertoa luvulla 10^{-2}. Syynä tämän esitysmuodon käyttöön on, että se mahdollistaa laskut hyvin pienillä luvuilla. Seikkaperäinen selvitys lukujen esittämisestä on luvussa MATEMAATTISIA LASKUJA.

Helppolukuinen tulostus

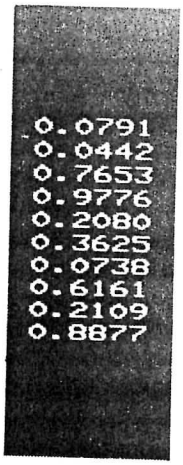
Jos haluat satunnaislukujen tulostuksen tavallisessa desimaalimuodossa, voidaan tämä helposti järjestää varustamalla käsky PRINT lisäyksellä, joka säätelee tulostuksen ulkomuotoa. Muuta ohjelmariviä 70 jollakin seuraavista tavoista:

```
70 PRINT USING "#.#######" RND
70 PRINT USING "#.####" RND
70 PRINT USING "#.##" RND
```

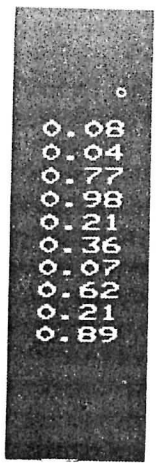
Kuten näet, voidaan kirjoitettavien numeroiden lukumäärään vaikuttaa.



USING "#.######"



USING "#.####"



USING "#.##"

Monissa yhteyksissä on välttämätöntä, että luvut voidaan esittää halutuilla desimaaleilla. Esimerkiksi markkojen ja pennien määrä kirjoitetaan aina kahdella desimaalilla. Käskyllä PRINT USING "..."... voidaan tulostukselle antaa toivottu muoto. Myöhemmin annamme lisää esimerkkejä käskyn PRINT USING "..."... käytöstä.

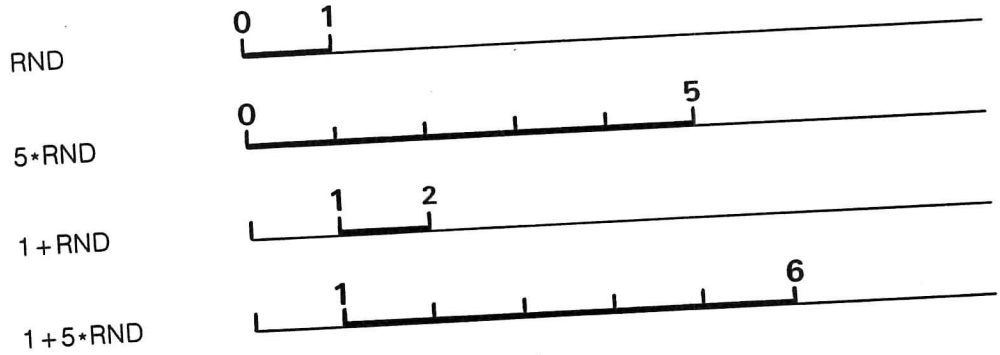
Satunnaislukuja toivotulta väliltä

Kuinka tietokone voi aikaansaada satunnaislukuja lukujen 0 ja 5 väliltä? Tämä tapahtuu kertomalla muodostetut satunnaisluvut 5:llä. Jos sen sijaan haluat käyttää satunnaislukuja lukujen 1 ja 2 väliltä, pääset tähän lisäämällä muodostettuihin satunnaislukuihin luvun 1.

Editoi peräkkäin edellisen ohjelman rivi 70 ja kokeile seuraavia vaihtoehtoja.

```
80 PRINT USING "#.# #" 5*RND
80 PRINT USING "#.# #" 1+RND
80 PRINT USING "#.# #" 1+5*RND
```

Kun jokaisen editoinnin jälkeen ajat ohjelman, voit todeta, että muodostettavat satunnaisluvut osuvat seuraaviin väleihin:



Mielivaltaisia satunnaislukuja

Kuten aikaisemmista ajoista olet huomannut, syntyy RND-funktiolla aina sama jono satunnaislukuja uutta ajoa suoritettaessa. Tämä ominaisuus on etu niin kauan kuin ohjelmaa edelleen koeajetaan. Sen jälkeen toivotaan luonnollisesti, että jokaisella uudella ajolla syntyy mielivaltaisia satunnaislukuja. Kun käsky RANDOMIZE suoritetaan, antaa RND-funktio täysin mielivaltaisia satunnaislukuja.

Lisää ohjelmarivi 45 ja tarkista, että joka ajolla muodostuu eri satunnaisluvut.

```
45 RANDOMIZE
```

Ohjelmasilmukoita uudella tavalla

Ohjelmajakso on joskus suoritettava määrätyn monta kertaa. Tämä voidaan tavalliseen tapaan suorittaa käskyjen WHILE ... ja WEND avulla. Vaihtoehtoinen tapa määritellä ohjelmasilmukka on käyttää hyväksi käskyjä FOR ... TO ... ja NEXT ... kuten seuraavassa ohjelmassa. Huomaa että ohjelmarivi 60 on poistettu.

```
10 PRINT CHR$(12)
20 PRINT "TIETOKONE ARPOO SATUNNAISLUKUJA"
30 PRINT "= = = = = = = = = = = = = = = = = = = = "
40 PRINT
45 RANDOMIZE
50 FOR N=1 TO 10
70  PRINT USING "#.# #" 1+5*RND
80 NEXT N
90 PRINT
```

Ohjelmarivi 50 määrittelee silmukan alun ja rivi 80 silmukan lopun. Joka kerran kun ohjelmajakso, ts. rivi 70, suoritetaan, lisääntyy silmukkamuuttujan (N) arvo 1:llä. Ohjelmariviä 70 suoritetaan siihen saakka, kunnes silmukkamuuttuja saavuttaa loppuarvonsa, tässä tapauksessa 10. Myös tämä ohjelma antaa siis tulokseksi 10 satunnaislukua.

Kun etukäteen tiedetään, kuinka monta kertaa ohjelmajaksoa aiotaan suorittaa, voi tämä tapa silmukan muodostamiseksi olla edullinen.

Noppapeli

Nopanheiton mukaileminen ei tietenkään ole todellinen tietokonetehtävä, mutta se voi toimia esimerkkinä satunnaislukujen käytöstä. Toivomme satunnaislukuja, jotka ovat kokonaislukuja ja jotka ovat välillä 1:stä 6:een. Tämä ohjelma esittelee viiden nopan heittoa ja se sisältyy myös osana noppapeliin nimeltä YATZY.

```
100 PRINT CHR$(12)
110 PRINT "N O P P A P E L I"
120 PRINT "= = = = = = = = = ="
130 PRINT CUR(6,0) "NOPAT OSOITTAVAT:"
140 PRINT
150 RANDOMIZE
160 FOR N=1 TO 5
170  PRINT INT(1+RND*6);
180 NEXT N
190 PRINT CUR(18,0);
200 INPUT "HALUATKO JATKAA? (KYLLÄ/EI) "S$
210 IF S$<>"EI" THEN GOTO 100
220 PRINT "AJO SUORITETTU"
```

Funktion INT (...) avulla valitaan suluissa olevan luvun kokonaisosa. Esimerkiksi luvun 6,18372 kokonaisosa on 6, ts. INT (6.18372) antaa tulokseksi 6.

Koska PRINT-lause ohjelmarivillä 170 loppuu puolipisteeseen, tapahtuu viiden satunnaisluvun tulostus kuvaruudun samalle riville.

Tietokone on liian nopea?

Tärkeää osaa noppapelissä näyttelee tuloksen odottelu. Edellisessä ohjelma-ajossa ilmestyivät kaikki silmäluvut välittömästi kuvaruudulle. Lisäämällä monta kierrosta käsittävän silmukan voimme muuttaa ohjelmaa siten, että ajo vie pitemmän ajan. Lisää tätä varten rivit:

```
164 FOR K=1 TO 1000
166 NEXT K
```

Näiden 1000 kierroksen suorittaminen aikaansaa noin yhden sekunnin viivytyksen. Ellet halua vakiopituista viivettä ennen kunkin silmäluvun tulostusta, voit muuttaa viivesilmukkaa esim. seuraavalla tavalla:

```
164 FOR K=1 TO 3000*RND
```

Koska silmukkamuuttujan loppuarvo vaihtelee sattumanvaraisesti, tulee viiveistä 0—3 sekunnin mittaisia.

Automaattinen pankki

Nykyaikana on mahdollista hoitaa erilaisia pankkiasioita pankin ollessa suljettuna. Esimerkiksi tililtä voidaan nostaa pankkiautomaatin avulla. Tämä on mahdollista sen vuoksi, että tietokoneet hoitavat kaiken pankkiautomaattiin liittyvän rekisteröinnin.

Tämän luvun viimeisenä esimerkkinä tutustumme ohjelmaan, joka näyttää miten pankkiautomaatti toimii. Olennaisia yksinkeraistuksia on tietenkin pakko tehdä, sillä mm. kortinlukija sekä kuitinpainaja puuttuu. Oletamme että pankkiautomaattimme täyttää seuraavat edellytykset:

- Pankissa on vain yksi tili.
- Tilin saldo on 10 000 mk, kun ohjelman ajo alkaa.
- 'Salaisena' koodina on RAHA.
- Otto ei saa ylittää senhetkistä saldoa.
- Jokaisen oton jälkeen tulostuu kuvaruudulle 'kuitti'. Sen tulee olla näkyvillä 15 sekuntia ja sen tulee näyttää seuraavalta:

Ohjelma, joka toteuttaa käytännössä 'pankkiautomaattimme', voi näyttää seuraavalta.

```
100 LET S=10000
110 PRINT CHR$(12)
120 PRINT CUR(4,0) "TERVETULOA !"
130 INPUT "ANNA SALAINEN KOODI: "S$
140 IF S$<>"RAHA" THEN GOTO 130
150 INPUT "MIKÄ OTTO ?        "B
160 IF B>S THEN LET B=S
170 LET S=S—B

180 PRINT CHR$(12)
190 PRINT CUR(4,0) "PÄIVÄMÄÄRÄ:      AIKA:"
200 PRINT TIME$
210 PRINT CUR(7,0) "OTETTU SUMMA:    " B
220 PRINT "TILILLÄ JÄLJELLÄ: " S
230 PRINT CUR(11,0) "TERVETULOA UUDELLEEN!"
240 FOR N=1 TO 15000
250 NEXT N

260 GOTO 110
```

Jotta ohjelmaesimerkki olisi helpompi ymmärtää, selvitellään seuraavaksi rivi riviltä ohjelman sisältöä.

Rivinumero	**Sisältö**
100	Muuttujalle S annetaan arvoksi saldo 10 000.
110, 180	Kuvaruutu tyhjennetään.

Rivinumero	Sisältö
120	Tulostus riville 4 alkaen sarakkeesta 0.
130	Alkuteksti sekä muuttujan S-ọ- koodiarvon syöttö näppäimistöltä.
140	Virheellinen koodi aiheuttaa uuden syöttöpyynnön.
150	Alkutekstin tulostus sekä muuttujan B arvon syöttö.
160	Jos arvo on suurempi kuin saldo, muutetaan arvo saldon suuruiseksi.
170	Oton jälkeisen uuden saldon laskeminen.
190, 210 220, 230	Selittävän tekstin ja muuttujan arvon tulostus annetulle paikalle.
200	Funktion TIME-ọ- arvo on kulumassa oleva ajanhetki. Katso tarkemmin seuraavalta sivulta.
240, 250	Silmukka joka aikaansaa noin 15 sekunnin viiveen.
260	Ehdoton hyppy ohjelmariville 110. Ohjelma tekee ikuista silmukkaa ja ajo voidaan lopettaa ainoastaan kaksinkertaisella painalluksella näppäimiin CTRL ja C

Huom! Komennolla LIST ilmestyvät ainoastaan ohjelman 20 ensimmäistä riviä kuvaruudulle. Jokaisella välilyöntitangon painalluksella saadaan yksi lisärivi näkyviin.

Tietokoneen kellon asettaminen

Tietokoneessa on sisäinen kello, joka toimii riippumatta tietokoneen muista toiminnoista. Kello antaa tiedon vuodesta, kuukaudesta ja päivästä sekä tunneista, minuuteista ja sekunneista. Kello voidaan asettaa joko ohjelma-ajon aikana tai komennon avulla. Kellon luku tapahtuu käyttämällä funktiota TIME-ϕ-. Asettaaksesi esim. ajan 1983-12-22 1.30.0, on Sinun annettava komento:

```
POKE -17,83,12,22,1,30,0
```

Komento lopetetaan painamalla näppäintä RETURN täsmälleen asetettavalla kellonlyömällä.

Yhteenveto

Käskyjä

FOR ... TO ... STEP ...

NEXT ...

Toistaa ohjelmanosan niin monta kertaa kuin silmukkamuuttujan alku- ja loppuarvo sekä askelarvo määräävät.

Jos askelarvo jätetään pois, se asetetaan arvoon 1.

```
20 FOR K = 1 TO 20 STEP 2

40 NEXT K
```

PRINT USING "..."...

Ilmoitetun lauseen tulostus esitetään halutussa muodossa. Ks. käyttöohjetta.

```
20 PRINT USING "+ # # # .# #"  A;B
```

RANDOMIZE

Antaa RND-funktiolle satunnaisen alkuarvon.

Lausekkeet ja operaattorit

Lauseke koostuu luvuista ja funktioista, joita erottavat operaattorit. Seuraavat operaattorit ovat olemassa:

PRIORITEETTI	OPERAATTORI	MERKITYS	ESIMERKKI
1	Ü (vaiht.**)	eksponentti	AÜB
2	*	kertolasku	A*B
2	/	jakolasku	A/B
3	+	yhteenlasku	A+B
3	–	vähennyslasku	A–B
4	>	suurempi kuin	A>B
4	<	pienempi kuin	A<B
4	=	yhtä suuri kuin	A=B
4	>=	suurempi tai yhtä suuri kuin	A>=B
4	<=	pienempi tai yhtä suuri kuin	A<=B
4	<>	eri suuri kuin	A<>B

Vertailuoperaattorit, joiden prioriteetti on 4, antavat arvon TOSI (–1), jos lauseke on tosi, muutoin arvon EPÄTOSI (0). Vrt. LOOGISET OPERAATTORIT.

Matemaattiset funktiot

INT (X) suurin kokonaisluku, joka on pienempi tai yhtä suuri kuin x.

RND esittää satunnaisluvun väliltä 0—0.999999.

Erityisfunktiot

CUR (R,K)

Siirtää kohdistimen (kursorin) riville R sarakkeeseen K kuvaruudulla. Esiintyy ainoastaan PRINT-lauseissa.

$0 \leqslant R \leqslant 23$ (rivit) $0 \leqslant K \leqslant 39/79$ (sarakkeet)

```
20 PRINT CUR (0,26) "OIKEALLA YLHÄÄLLÄ"
```

TIME-◊-

Antaa sisäänrakennetun kellon päiväyksen ja kellonajan.

```
10 POKE –17, Y, M, D, H, M1, S! Asetus
20 PRINT TIME-◊-
```

KOMENNOT

AUTO ...,...
Aiheuttaa automaattisen rivinumeroinnin syöttämisen yhteydessä.

AUTO	(rivinumero: 10, intervalli: 10)
AUTO 100	(rivinumero: 100, intervalli: 10)
AUTO 500,50	(rivinumero: 500, intervalli: 50)

ED ...
Mahdollistaa ohjelmarivin muuttamisen ilman uudelleen kirjoittamista. Näppäin kopioi yhden merkin ja vastaavasti näppäin poistaa merkin.
ED 70

REN ...,...,...—...
(vaiht. RENUMBER ...,...,...—...)
Ilmoitetuille riveille annetaan uudet rivinumerot. Seuraavat muunnokset ovat olemassa:
Seuraavat esimerkit koskevat kaikkia ohjelmarivejä:

REN	(ens. rivinumero: 10, intervalli: 10)
REN 100	(ens. rivinumero: 100, intervalli: 10)
REN 100,20	(ens. rivinumero: 100, intervalli: 20)

Seuraavat esimerkit koskevat ilmoitettua ohjelmaosuutta, jonka ensimmäinen rivinumero on 600 ja intervalli 10:

REN 600,10,500—800	(rivinumerot 500—800)
REN 600,10,—800	(rivinumerot riviin 800 saakka)
REN 600,10,500—	(rivinumerot alkaen rivistä 500)

Harjoitustehtäviä

* 4.1 Muuta sivun 57 ohjelmaa siten, että se laskee kahden luvun osamäärän.

* 4.2 Laadi ohjelma, joka laskee alennetun hinnan (H), kun alkuperäinen hinta (A) ja alennusprosentti (P) annetaan. Alennettu hinta lasketaan kaavalla H = A — A * P / 100.

* 4.3 Laadi ohjelma, jota voit käyttää shekkitilisi saldon tarkkailuun. Syöttötietoina annetaan saldo, josta laskenta aloitetaan, sekä tapahtuma. Jokaisen uuden syötön jälkeen lasketaan ja tulostetaan uusi saldo.

* 4.4 Palkkaneuvottelija haluaa laskea 'käteen jäävän' tuntipalkan korotuksen, kun kuukausipalkka nousee tietyn määrän. Kirjoita ohjelma, joka pyytää kuukausipalkan korotuksen (L), marginaaliveroprosentin (M) ja kuukauden työtuntien määrän (T). Ohjelman tulee tämän jälkeen laskea ja ilmoittaa tuntipalkan korotus veronpidätyksen jälkeen.
Vihje: Tuntipalkan korotus = (L − M * L / 100) / T.

* 4.5 Kun K markan suuruinen pääoma kasvaa P prosentin mukaan korkoa vuoden, on kasvanut pääoma laskettavissa lausekkeesta K + K * P / 100. Laadi ohjelma, joka laskee kasvaneen pääoman vuosi vuodelta. Syöttötietoina ovat alkuperäinen pääoma ja korkoprosentti.

* 4.6 Liike myöntää paljousalennusta seuraavasti:
10 %, jos ostoksien arvo vähintään 50 mk,
20 %, jos ostoksien arvo vähintään 200 mk,
30 %, jos ostoksien arvo vähintään 500 mk ja
40 %, jos ostoksien arvo vähintään 1000 mk.
Laadi ohjelma, joka laskee ostoksien alennetun hinnan, kun alentamaton hinta annetaan.

* 4.7 Laadi ohjelma, joka arpoo veikkausrivin. Oletetaan, että ykkösen todennäköisyys on 60 %, ristin 20 % ja kakkosen 20 %.

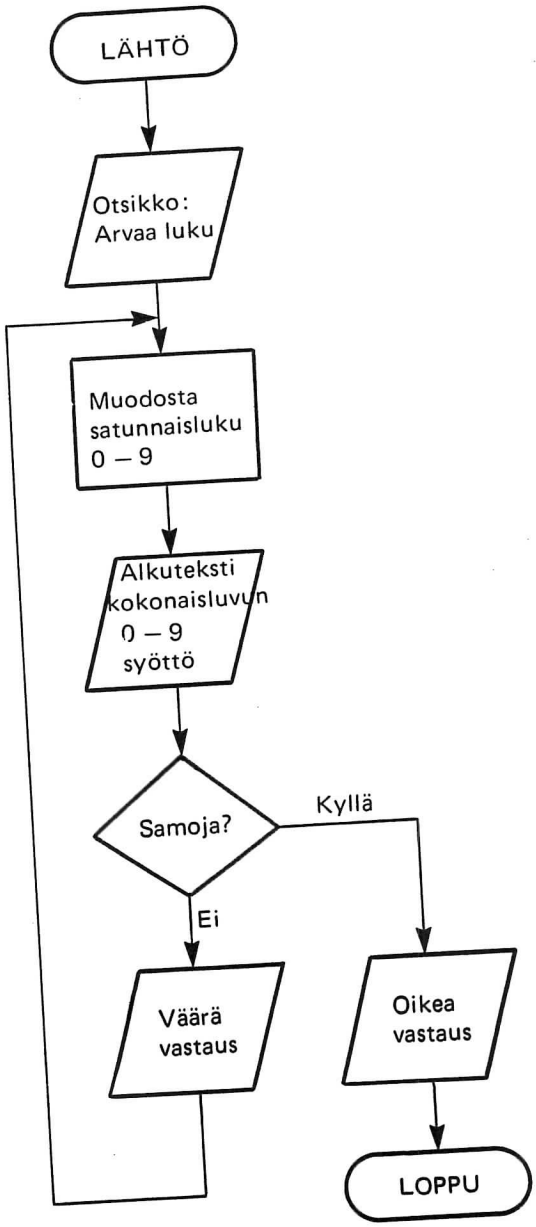
* 4.8 Laadi ohjelma, joka arpoo yhden lottorivin (7 eri kokonaislukua väliltä 1—37).

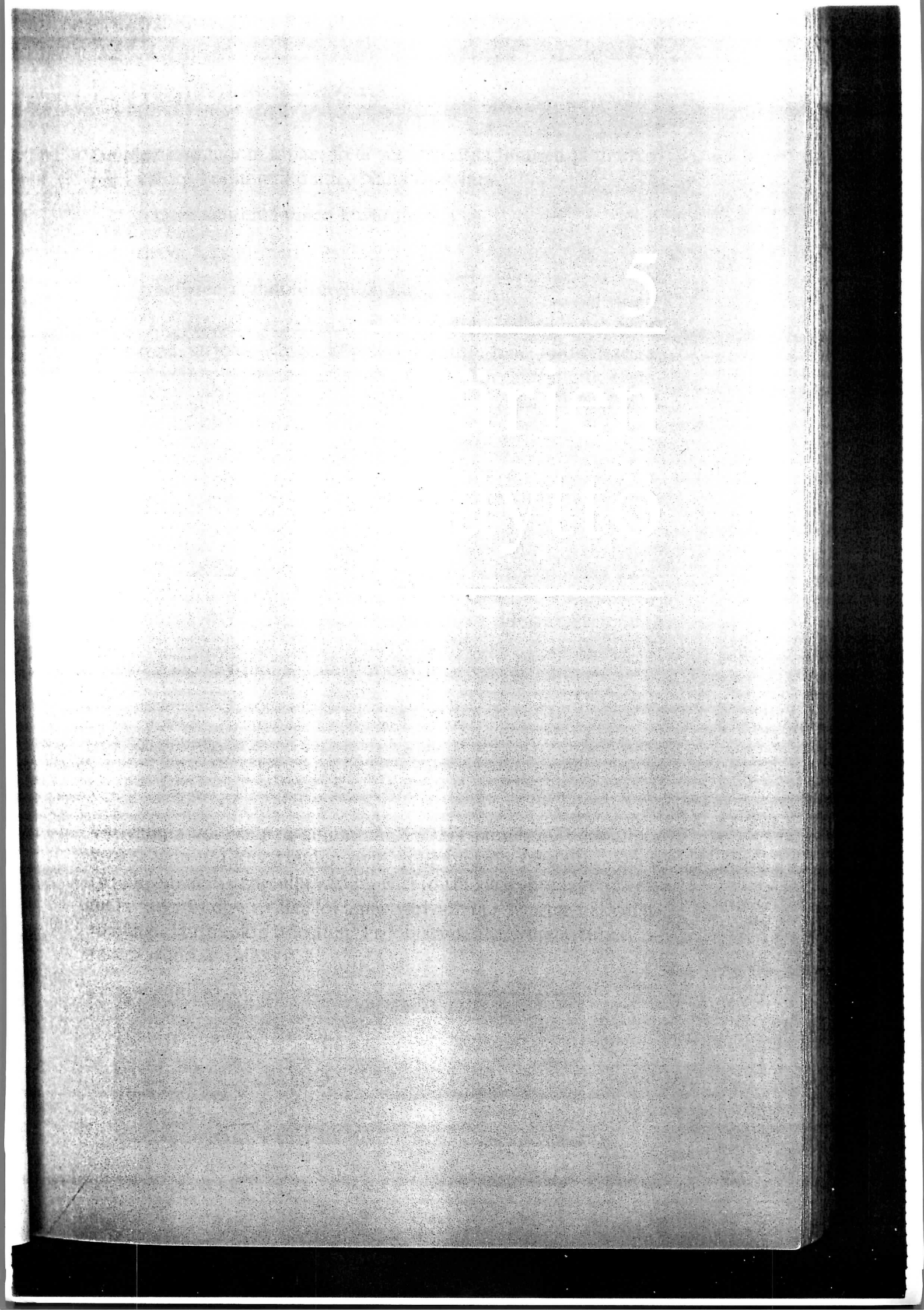
* 4.9 Seuraavaa ohjelmaa voidaan käyttää kellonajan jatkuvaan näyttämiseen kuvaruudulla:

```
210 PRINT CHR$(12)
220 PRINT CUR(0,20) TIME$
230 GOTO 220
```

Kuten huomaat, ohjelma tekee ikisilmukkaa ja se voidaan lopettaa vain tietokoneen hätäkatkaisimella. Täydennä ohjelmaa siten, että kello voidaan asettaa aikaan. Syöttötietoina ovat vuosi, kuukausi, päivä, tunti, minuutti ja sekunti.

* 4.10 Vuokaavio esittää ohjelmaa, jossa tietokone muodostaa satunnaislukuja 0:sta 9:ään. Kun annat arvauksesi, vastaa tietokone OIKEA VASTAUS tai VÄÄRÄ VASTAUS. Laadi vuokaaviota vastaava ohjelma.





Esitys parhaalla tavalla

On olemassa monta tapaa, jolla voi helpottaa ihmisen ja tietokoneen välistä kommunikointia. Näitä ovat mm.

- tekstin suurentaminen kuvaruudulla
- tietyn tekstin merkitseminen värillä
- graafisten mahdollisuuksien käyttö.

Tässä kappaleessa esitämme, mitä mahdollisuuksia ABC 800C -tietokone tarjoaa värien käytössä kuvaruudulle tulostettaessa. Esimerkkejä graafisista merkeistä on tässä luvussa, mutta enemmän tietoa graafisista sovellutuksista löytyy luvusta GRAAFISIA MAHDOLLISUUKSIA.

Värillinen teksti

ABC 800 -tietokoneella voidaan kirjoittaa kuutta eri väriä käyttäen. Tällöin täytyy signaalikaapelin RGB/GREEN -kytkimen olla asennossa RGB. Halutessasi tietyn värisen tulostuksen ilmoita tämä väri vastaavassa PRINT-lauseessa. Periaate, jolla tekstin väri asetetaan, käy ilmi seuraavista ohjelmariveistä.

```
10 PRINT "VALKOISET SIREENIT"
20 PRINT RED "PUNAISET RUUSUT"
30 PRINT BLU "SININEN MERI"
40 PRINT YEL "KESÄ JA AURINKO"
50 PRINT YEL "AURINKO" RED "RUUSU" BLU "MERI"
```

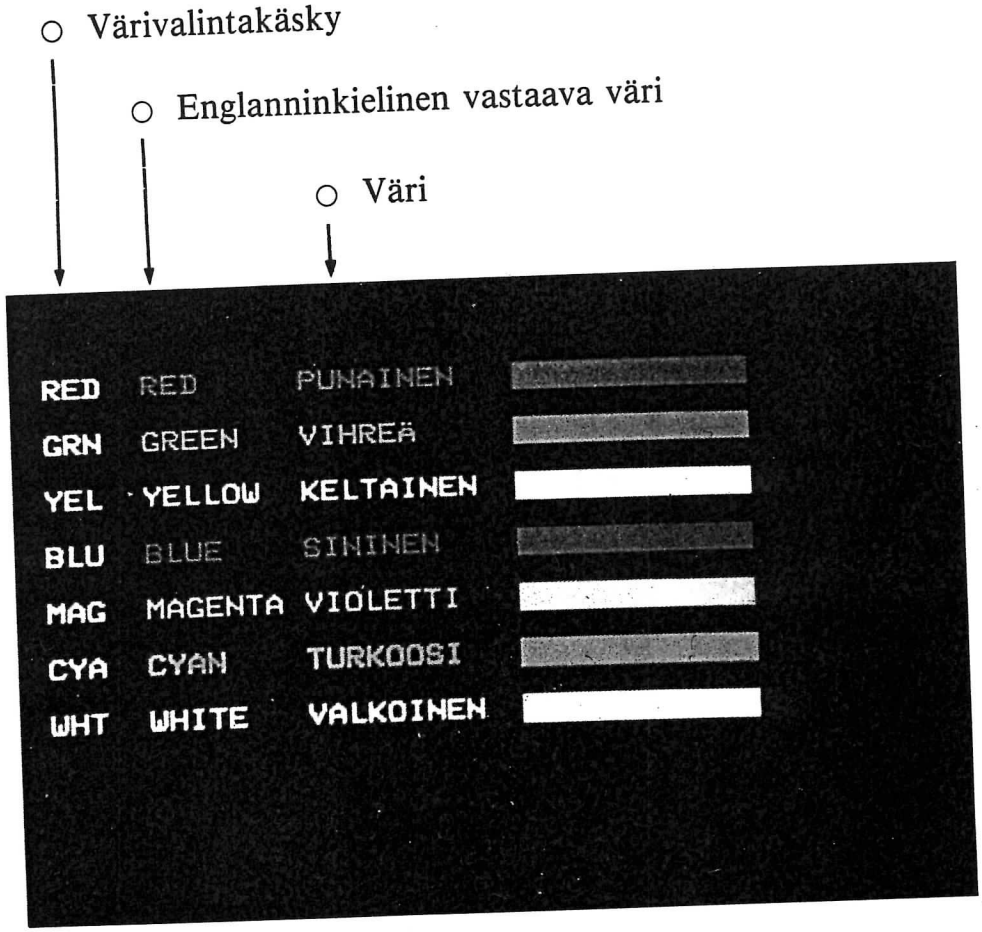
Värivalinta voi esiintyä ainoastaan PRINT-lauseissa. Värivalinta koskee vain sitä kuvaruudun riviä, jolle tulostus tapahtuu. Koko rivi kirjoitetaan annetulla värillä, ellei toista värivalintaa tehdä samalle kuvaruudun riville. Jokainen värivalinta aiheuttaa näkymättömän ohjausmerkin tulostumisen kuvaruudulle, kuten seuraavasta kuvaruudusta näkyy.

Näkymättömien ohjausmerkkien paikat kuvaruudulla näkyvät tässä kuvassa.

VALKOISET SIREENIT
□ PUNAISET RUUSUT
□ SININEN MERI
□ KESÄ JA AURINKO
□ AURINKO □ RUUSU □ MERI

Mitä värejä voi käyttää

Seuraava valokuva näyttää värivalintakäskyt ja vastaavat värit.



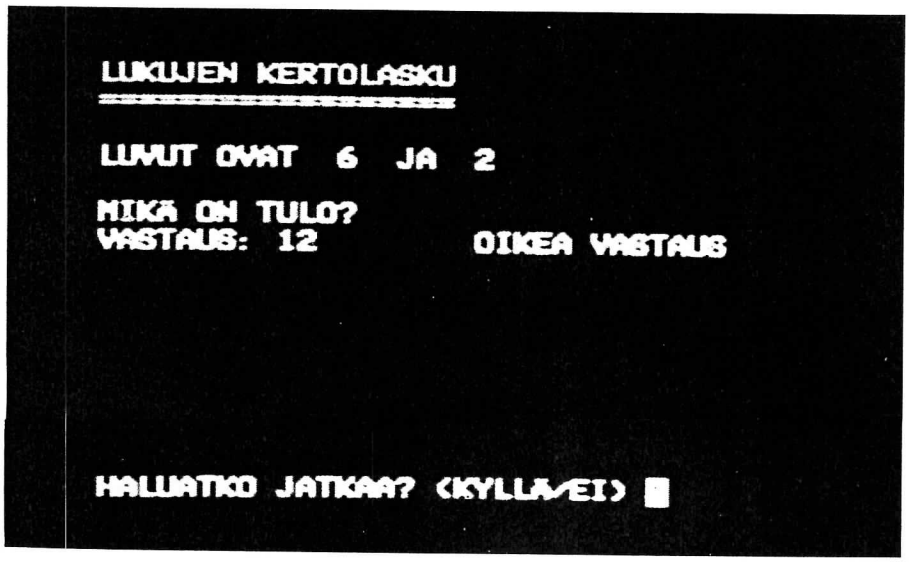
Kun värivalintakäsky puuttuu, tulostuu teksti valkoisena. Käskyä WHT käytetään vain silloin, kun värillisen rivin tekstistä osa halutaan tulostaa valkoisella.

Värillinen otsikko

Värillistä tekstin tulostamista käytetään tässä korostamaan otsikkoa. Alla olevassa esimerkissä on yksi muunnos kertolaskuohjelmasta, joka esitettiin luvussa 4. Esimerkkiä täydennetään nyt erilaisten värillisten tekstien avulla.

```
100 PRINT CHR$(12)
110 PRINT YEL "LUKUJEN KERTOLASKU"
120 PRINT YEL "= = = = = = = = = = = = = = = "
130 RANDOMIZE
140 LET A=INT(10*RND)
150 LET B=INT(10*RND)
160 PRINT CUR(4,1) "LUVUT OVAT " A " JA " B
170 PRINT CUR(6,1) "MIKÄ ON TULO?" CUR(7,1);
180 INPUT "VASTAUS: "V
190 PRINT CUR(7,20);
200 IF V=A*B THEN PRINT "OIKEA VASTAUS"
210 IF V<>A*B THEN PRINT "VÄÄRÄ VASTAUS!"
220 PRINT CUR(16,1);
230 INPUT "HALUATKO JATKAA? (KYLLÄ/EI) "V$
240 IF V$<>"EI" THEN GOTO 100
250 PRINT "AJO ON PÄÄTTYNYT"
```

Kun ohjelmarivit 110 ja 120 suoritetaan, tulostuu vastaaville riveille keltaisen värin ohjausmerkit ensimmäiseen sarakkeeseen. Tulostus rivien 160, 170, 180 ja 230 mukaan aloitetaan siksi sarakkeesta 1.



Jotta kiinnittäisimme huomiota kerrottaviin lukuihin ja vastauksen paikkaan, muutetaan hieman ohjelmarivejä 160 ja 170.

```
160 PRINT CUR(4,1) "LUVUT OVAT " YEL A WHT " JA " YEL B
170 PRINT CUR(6,1) "MIKÄ ON LUKUJEN TULO?" CUR(7,0) YEL;
```

Haluttaessa voidaan antaa sopivilla väreillä myös vaihtoehdot "OIKEA VASTAUS" ja "VÄÄRÄ VASTAUS!"

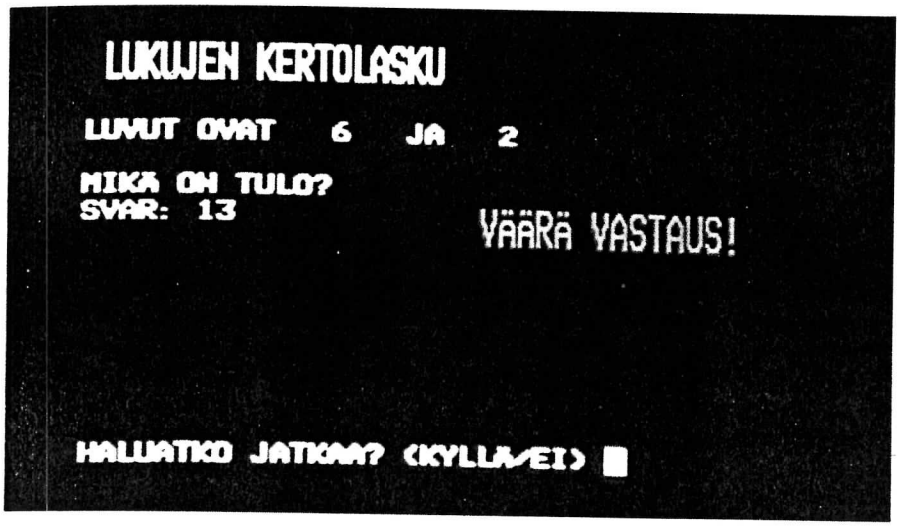
```
200 IF V=A*B THEN PRINT GRN "OIKEA VASTAUS"
210 IF V<>A*B THEN PRINT RED "VÄÄRÄ VASTAUS!"
```

Suurennettu teksti

Toinen tapa kiinnittää huomiota tiettyyn tekstinosaan on käyttää kaksi kertaa korkeampia kirjaimia. Tämä saadaan aikaan käskyllä DBLE, joka värivalintakäskyn tavoin voi esiintyä ainoastaan PRINT-lauseissa. Jos esimerkiksi otsikko ja ilmoitus "VÄÄRÄ VASTAUS!" halutaan kirjoittaa kaksinkertaisena (engl. DouBLE) muutetaan näitä rivejä:

```
110 PRINT DBLE YEL "LUKUJEN KERTOLASKU"
210 IF V<>A*B THEN PRINT DBLE "VÄÄRÄ VASTAUS!"
```

Kun otsikko tulostetaan tavallisen kokoisilla kirjaimilla, sijoittuu se riville 1. Kaksinkertaisilla kirjaimilla tulostettu otsikko varaa sekä rivin 1 että rivin 2. Sen vuoksi riviä 2 ei voi käyttää muuhun tulostukseen.



Kuten huomaat, ei ohjelmarivi 120 aiheuta kuvaruudulle mitään tulostusta. Jos tietylle kuvaruudun riville on kirjoitettu suurennettua tekstiä, voidaan sille taas kirjoittaa normaalikokoista tekstiä käyttämällä käskyä NRML (engl. NoRMaL). Käskyt DBLE ja NRML aiheuttavat näkymättömän ohjausmerkin tulostumisen kuvaruudulle samoin kuin värillistä tekstiä käytettäessä.

Taustavärin valinta

Tavallisesti kuvaruudulle tulostettaessa tausta on musta. Haluttaessa värillinen tausta käytetään käskyä NWBG (engl. NeW BackGround) tekstin värivalinnan jälkeen. Paluu mustaan taustaväriin tapahtuu käskyllä BLBG (engl. BLack BackGround).

Taustaväriin vaikuttava käsky aiheuttaa näkymättömän ohjausmerkin tulostumisen kuvaruudulle. Kuten tavallista, tämä käsky vaikuttaa ainoastaan ohjausmerkkiä seuraavaan kuvaruudun riviin.

Muuttakaamme edellistä esimerkkiohjelmaamme siten, että otsikko esitetään sinistä taustaväriä vasten. Ohjelmarivi 120 poistetaan ennen ohjelman ajoa.

```
110 PRINT BLU NWBG YEL "LUKUJEN KERTOLASKU "BLBG
```

Vilkkuva teksti

Esimerkiksi virheellisestä syöttötiedosta voisi olla mukavaa ilmoittaa käyttämällä vilkkuvaa tekstiä. Käskyllä FLSH (engl. FLaSH) saa kuvaruudun tulostettavan rivin vilkkumaan. Jos vain osan rivistä halutaan vilkkuvan, käytetään käskyä STDY (engl. STeaDY), joka lopettaa muun rivin vilkkumisen.

Esimerkissämme teksti "VÄÄRÄ VASTAUS!" tulostetaan vilkkuvana, kun ohjelmariville 210 tehdään seuraava muutos:

```
210 IF V<>A*B THEN PRINT DBLE RED FLSH "VÄÄRÄ VASTAUS!"
```

Jos kokeilet ohjelmaa omilla muunnoksilla, tyhjennä tietokoneen muisti komennolla NEW.

Graafisia mahdollisuuksia

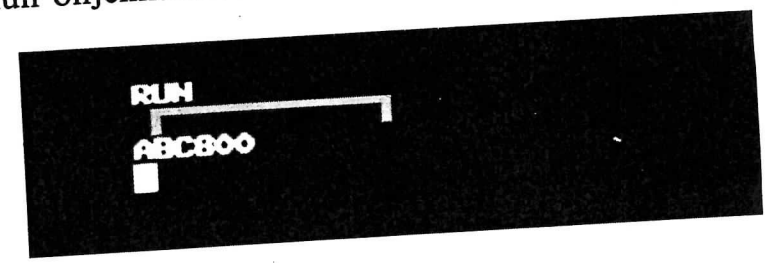
Hyvä kuva helpottaa aina esitetyn tiedon ymmärtämistä. Tämä johtuu mm. siitä, että ihmisen on helpompi havaita muutoksia kuvassa kuin kirjoitetussa tekstissä.

ABC800C -tietokoneella voi kuvia esittää monella eri tavalla. Tarkastelemme nyt yksinkertaisinta menetelmää eli kuvan esittämistä graafisia merkkejä käyttäen.

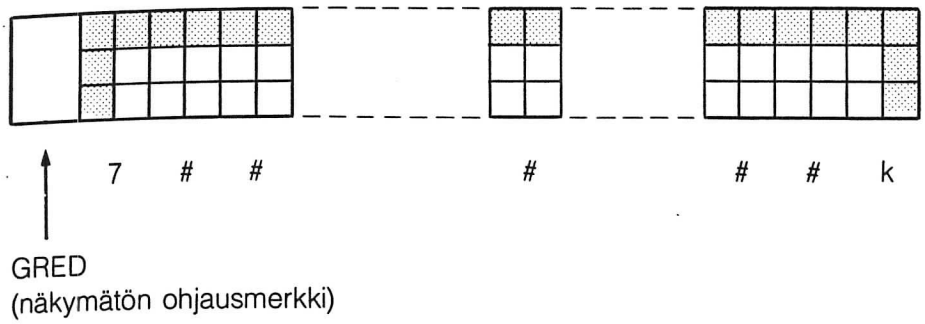
Graafiset merkit voidaan tulostaa kuvaruudulle halutun värisinä käyttämällä samantapaisia käskyjä kuin aiemmin käytettiin tekstin väriä valittaessa. Toivotun värisen grafiikan ohjausmerkki aiheuttaa PRINT-lauseessa esiintyvien aakkosnumeeristen merkkien tulostumisen kuvaruudulle graafisina merkkeinä.

```
10 PRINT GRED "7#########k"
```

Kun ohjelmarivi 10 suoritetaan, tulostuu seuraava kuva.



Periaate, jolla kuvat muodostetaan, näkyy tästä kuvaruudun riviä esittävästä kuvasta.



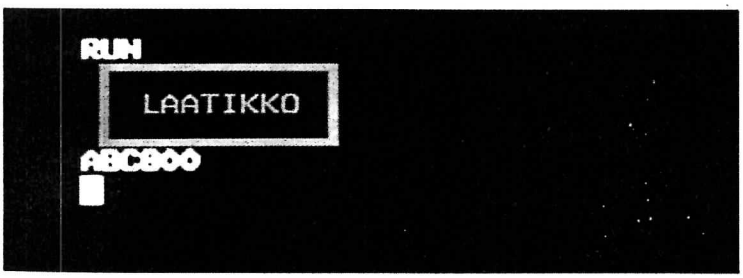
Grafiikkakäskyt ovat GRED, GGRN, GYEL, GBLU, GMAG, GCYA ja GWHT. Graafiset merkit voidaan tulostaa käyttämällä samaa väriä kuin tekstiä tulostettaessa.

Käytettävissä olevat graafiset merkit on lueteltu liitteenä olevassa taulukossa.

Taulukosta käy ilmi, että isot kirjaimet tulostuvat tavallisella tavalla, vaikka kuvaruudun rivi olisikin ''grafiikkatilassa''. Kokeile esimerkiksi ohjelmarivejä:

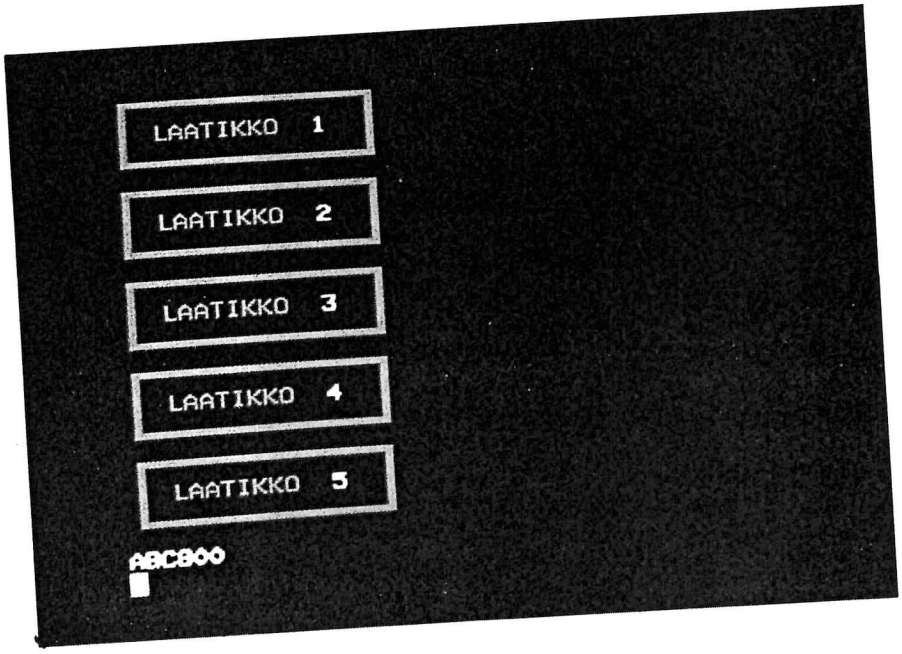
```
20 PRINT GRED "5 LAATIKKO j"
30 PRINT GRED "upppppppppz"
```

Ajettaessa ohjelma tulostuu kuvaruudulle seuraava kuva.



Ohjelmarivi, joka on aikaisemmin asetettu ''grafiikkatilaan'', saadaan muutetuksi ''kirjaintilaan'' käyttämällä apuna värivalintakäskyä. Tällä tavalla voidaan graafisia ja aakkosnumeerisia merkkejä sekoittaa halutulla tavalla. Seuraavalla sivulla olevassa esimerkissä käytetään tätä tekniikkaa ohjelmarivillä 20.

```
2 PRINT CHR$(12)
5 FOR K=1 TO 5
10  PRINT GRED "7# # # # # # # # # #k"
20  PRINT GRED "5 LAATIKKO" YEL K GRED "j"
30  PRINT GRED "upppppppppppz"
40  PRINT
50 NEXT K
```



Prosessin kuvaus

Prosessiteollisuudessa on prosessin etenemistä tapana kuvata värillisenä kuvaruudulla. Tämä valokuva näyttää, miten ABC800C -tietokoneen värigrafiikkaa voidaan käyttää hyväksi tällaisissa yhteyksissä.

Ohjelma, joka muodostaa tämän kuvan, on tällainen:

```
100 PRINT CHR$(12)
110 INPUT "ANNA PÄIVÄYS JA KELLOAIKA (V,K,P,T,M,S)"V,K,P,T,M,S
120 POKE -17,V,K,P,T,M,S
130 PRINT CHR$(12)
140 PRINT DBLE YEL "P R O S E S S I"
150 PRINT CUR(7,0);
160 PRINT GYEL "■■■■■■■■"
170 PRINT GYEL "■■■■■■■■"
180 PRINT GYEL "■■■■■■■■"
190 PRINT GBLU "■■■■■■■■" GYEL "          ■■■■■■"
200 PRINT GBLU "                        ■■■■■■"
210 PRINT CUR(13,0) RED "SÄILIÖ          TANKKI"
220 PRINT CUR(11,6) "j00000000000p"
230 FOR K=1 TO 300 : NEXT K
240 PRINT CUR(11,6) "jéééééééééép"
250 FOR K=1 TO 300 : NEXT K
260 PRINT CUR(1,19) NRML WHT TIME$;
270 GOTO 220
```

Tarkastelemme nyt ohjelman toimintaa rivi riviltä.

Rivinumero	**Rivin merkitys**
100	Kuvaruudun tyhjennys.
110—120	Kellon asetus.
130	Kuvaruudun tyhjennys.
140	Otsikko kirjoitetaan kaksinkertaisella tekstikorkeudella ja keltaisella värillä.
150	Kohdistin siirretään riville 7 sarakkeeseen 0.
160—200	Kuvio muodostetaan graafisilla merkeillä.
210	Selittävä teksti tulostetaan.
220, 240	Erilaisia graafisia kuvioita esitetään vuorotellen, jotta saadaan aikaan liikkeen vaikutelma.
230	Silmukka, joka aiheuttaa n. 0,3 sekunnin viivytyksen. Huomaa, että kaksi käskyä voidaan kirjoittaa samalle riville kaksoispisteellä erotettuna.

250 Viivytyssilmukka.

260 Aika tulostetaan kuvaruudun oikeaan yläkulmaan.

270 Ehdoton hyppy ohjelmariville 220.

Suosittelemme sinua muuttelemaan ohjelmaa omien ajatustesi mukaan.

Väri ja grafiikka teksti-TV:ssä

Yksi esimerkki värillisen tekstin ja yksinkertaisten graafisten kuvioiden käytöstä tiedonvälityksessä on Teksti-TV.

Teksti-TV:n grafiikka on samanlainen kuin ABC800C -tietokoneen yksinkertaisin grafiikka. ABC800C:n graafisia mahdollisuuksia käsitellään laajemmin myöhemmin tässä kirjassa.

Värillisen tekstin käytön esittelyn lopetamme tähän. Sinun, jolla on mahdollisuus käyttää ABC800C -tietokonetta, täytyy tietysti jatkossakin käyttää hyväksesi värillisen tekstin suomia etuja.

Yhteenveto

Väri ja grafiikka ABC800C:lle

Huomaa, että seuraavat erikoiskäskyt väriä ja grafiikkaa varten esiintyvät vain PRINT-lauseissa.

RED GRN YEL BLU MAG CYA WHT
Tulostaa tavalliset merkit värillisinä (merkkitila).

```
10 PRINT YEL "aurinko" RED "ruusu" BLU "meri"
```

GRED GGRN GYEL GBLU GMAG GCYA GWHT
Tulostaa grafiikkamerkit värillisinä (grafiikkatila).

```
20 PRINT GGRN "wä" GCYA "'wä"
```

FLHS STDY
Antaa vilkkuvan ja normaalin tulostuksen.

```
30 PRINT FLSH YEL "tähti"
```

DBLE NRML
Tulostaa kaksinkertaisella ja normaalilla merkkikorkeudella.

```
40 PRINT RED DBLE "TAULUKKO"
```

NWBG BLBG
Muuttaa rivin taustavärin ja palauttaa sen normaaliksi (mustaksi). Haluttu taustaväri annetaan ennen NWBG-käskyä.

```
50 PRINT BLU NWBG YEL "AURINKO" BLBG
```

ASCII-merkit merkkitilassa (M) ja grafiikkatilassa (G)

Näppäinkoodit merkki/grafiikkamuodossa (ASCII-taulukko)

A	M	G	A	M	G	A	M	G	A	M	G
32	Tyhjä		56	8		80	P	P	104	h	
33	!		57	9		81	Q	Q	105	i	
34	"		58	:		82	R	R	106	j	
35	#		59	;		83	S	S	107	k	
36	¤		60	<		84	T	T	108	l	
37	%		61	=		85	U	U	109	m	
38	&		62	>		86	V	V	110	n	
39	'		63	?		87	W	W	111	o	
40	(		64	É	É	88	X	X	112	p	
41	)		65	A	A	89	Y	Y	113	q	
42	*		66	B	B	90	Z	Z	114	r	
43	+		67	C	C	91	Ä	Ä	115	s	
44	,		68	D	D	92	Ö	Ö	116	t	
45	-		69	E	E	93	Å	Å	117	u	
46	.		70	F	F	94	Ü	Ü	118	v	
47	/		71	G	G	95	-	-	119	w	
48	0		72	H	H	96	é		120	x	
49	1		73	I	I	97	a		121	y	
50	2		74	J	J	98	b		122	z	
51	3		75	K	K	99	c		123	ä	
52	4		76	L	L	100	d		124	ö	
53	5		77	M	M	101	e		125	å	
54	6		78	N	N	102	f		126	ü	
55	7		79	O	O	103	g		127	□	

ASCII-koodit (A) tulkittuina merkki-(M) ja grafiikkamuodossa (G).
Grafiikkamuoto kuuluu vain ABC 800 C-malliin.

CTRL ≧ Antaa merkin □ (neliö)

Kotkan
Kaupunginkirjasto

Harjoitustehtäviä

* 5.1 Muuta allaolevaa ohjelmariviä siten, että teksti esitetään

```
10 PRINT "VÄRILLINEN TULOSTUS"
```

a) punaisella värillä,

b) kaksinkertaisilla kirjaimilla punaisena,

c) punaisena ja vilkkuvana kaksinkertaisilla kirjaimilla ja

d) keltaista taustaa vasten.

* 5.2 Muuta sivun 62 esimerkkiohjelmaa siten, että otsikko kirjoitetaan kaksinkertaisilla kirjaimilla keltaisena. Esitä myös teksti "ARPAKUUTIOT OSOITTAVAT" keltaisena ja arpakuutioiden heiton tulokset kaksinkertaisella korkeudella ja punaisena.

* 5.3 Kun ohjelma "KOIRAKILPAILU" suoritetaan, näyttää siltä, että punainen koira useimmiten voittaa kilpajuoksun. Muuta ohjelmaa siten, että kumpikin koira voi voittaa yhtä suurella todennäköisyydellä.

```
100 PRINT CHR$(12)
110 RANDOMIZE
120 PRINT DBLE YEL "KOIRIEN KILPAJUOKSU"
130 A=A+RND*.4
140 B=B+RND*.3
150 PRINT CUR(5,A) GRED " i>"
160 PRINT CUR(7,B) GCYA " i>"
170 IF A>20 THEN PRINT CUR(5,25) RED FLSH "VOITTAJA nr 1"
180 IF B>20 THEN PRINT CUR67,25) CYA FLSH "VOITTAJA nr 2"
190 IF A>20 THEN GOTO 220
200 IF B>20 THEN GOTO 220
210 GOTO 130
220 PRINT CUR(20,0);
```

* 5.4 Eräs kauppias ilmoittaa "PÄIVÄN ERIKOISTARJOUKSET" värillisellä kuvaruudulla seuraavaa ohjelmaa käyttäen:

```
100 PRINT CHR-ϕ-(12)
110 PRINT CUR(3,2) DBLE FLSH YEL "PÄIVÄN ERIKOISTARJOUS:"
120 PRINT CUR(5,2) DBLE FLSH YEL "= = = = = = = = = = = = = = ="
130 PRINT CUR(10,12) DBLE YEL "MAKKARA  30,—/KG"
140 PRINT CUR(13,12) DBLE YEL "JUUSTO   25,—/KG"
150 PRINT CUR(16,12) DBLE YEL "SILLI    39,—/KG"
```

Muuta ohjelmaa siten, että teksti esitetään sinistä taustaa vasten.

5.5 Muuta ohjelmaa harjoitustehtävässä 4.3 siten, että taulukon otsikko ja arvot esitetään värillisinä.

5.6 On osoittautunut, että Kaustisten musiikkijuhlien sävellyskilpailussa pistemäärät jakautuvat aivan satunnaisesti eivätkä musiikillisten ansioiden mukaan. Tämän vuoksi päätettiin säästää ja korvata suuri tuomaristo tietokoneella. Seuraava ohjelma jakaa kullekin 10 kilpailukappaleesta satunnaisen pistemäärän välillä 0:sta 9:ään.

```
100 PRINT CHR$(12)
110 FOR N=1 TO 10
120  PRINT CHR-ϕ-(12)
130 .PRINT CUR(4,1) DBLE MAG "S Ä V E L M Ä  N R O " N
140  PRINT CUR(12,1) DBLE YEL ". . . . . "
150  PRINT CUR(14,1) DBLE YEL "0 1 2 3 4 5 6 7 8 9 "
160  PRINT CUR(18,1);
170  RANDOMIZE
180  K=INT(10*RND)
190  PRINT CUR(10,K*3) DBLE RED " "
200  PRINT CUR(20,2) CYA "SEURAAVA SÄVELMÄ (RETURN)";
210  INPUT V$
220 NEXT N
```

Muuta ohjelmaa siten, että pistemäärä voi vaihdella 0:sta 5:een ja että pistemäärä esitetään kuvaruudulla.

5.7 Seuraavassa ohjelmassa esiintyy käsky GSEP rivillä 130. Ajaessasi ohjelman voit todeta aiheuttaneesi graafisten merkkien muuttumisen (erottumisen).

```
100 PRINT CHR$(12)
110 FOR N=1 TO 10
120  PRINT CUR(N,0) GYEL "■■■■■■■■■■■■■■■■■■■■"
130  PRINT CUR(N,20) GYEL GSEP "■■■■■■■■■■■■■■■■■■■■"
140 NEXT N
```

5.8 Käsky GHOL antaa mahdollisuuden tulostaa kuvaruudulle edellinen graafinen merkki näkymättömän ohjausmerkin sijasta. Käskyn GHOL toiminta selviää seuraavasta ohjelmasta:

```
10 PRINT CHR$(12)
20 PRINT GYEL "■■" GHOL GBLU "■■" GHOL GYEL "■■■■■■"
30 PRINT GYEL "■■" GHOL GBLU "■■" GHOL GYEL "■■■■■■"
40 PRINT GBLU "■■■■■■■■■■■■"
50 PRINT GYEL "■■" GHOL GBLU "■■" GHOL GYEL "■■■■■■"
60 PRINT GYEL "■■" GHOL GBLU "■■" GHOL GYEL "■■■■■■"
70 FOR I=1 TO 12
80  PRINT GWHT "5"
90 NEXT I
100 PRINT CUR(18,0)
```

6

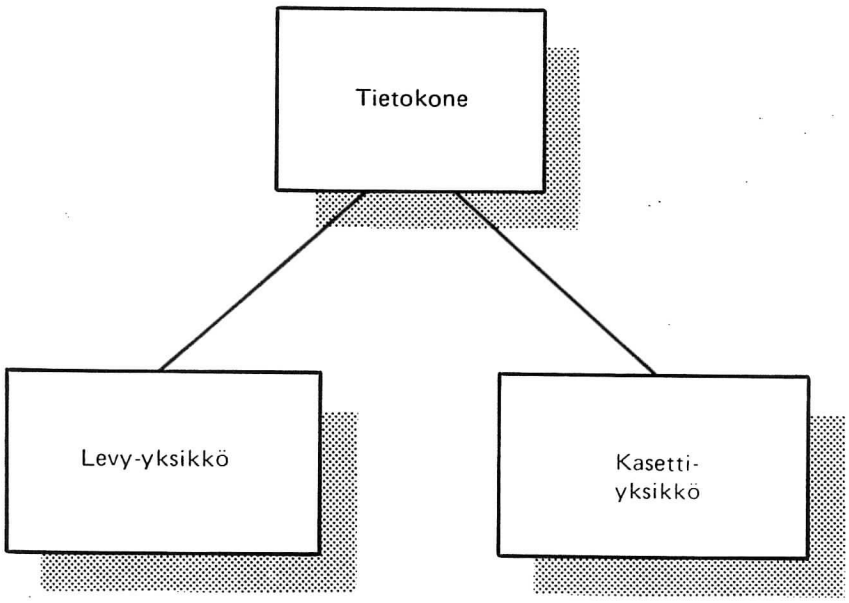
Miten vaimiita ohjelmia talletetaan?

Ohjelmien talletus ulkoiseen muistiin

Tietokoneen muistissa oleva ohjelma pysyy siellä ainoastaan niin kauan kuin tietokone on toiminnassa. Siksi kaikissa tietokonejärjestelmissä on olemassa mahdollisuus tallettaa ohjelmia magneettiselle laitteelle, jolla tieto säilyy, vaikka tietokone sammutettaisiinkin. Magneettiselle laitteelle talletettu ohjelma voidaan koska hyvänsä nopeasti ladata tietokoneen sisäiseen muistiin.

Tietokonejärjestelmän erilaisia apumuisteja

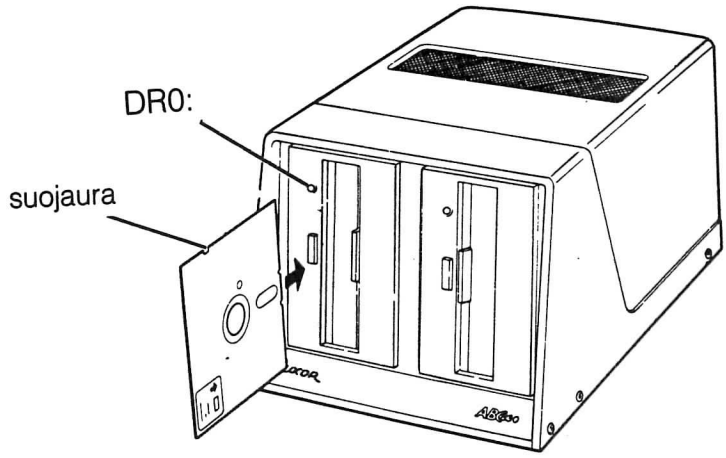
Useimpiin tietokonejärjestelmiin on olemassa erilaisia ulkoisia muistilaitteita ohjelmien ja datan tallettamista varten. Käsittelemme tässä kasettinauhuria ja levykeasemaa.



Tässä luvussa tarkastellaan, miten ohjelmia voidaan siirtää tietokoneen ja siihen kytketyn ulkoisen muistin välillä.

Ohjelman talletus levykkeelle

Kytke levykeasema käyttöohjeiden mukaisesti ja aseta sitten levyke (*) vasempaan levykeyksikköön (DR0:) kuvan mukaisesti. Oletamme tässä, että levyke on alustettu käyttöä varten. Levykkeen alustaminen opetetaan tämän luvun lopussa.



((*) Levykkeistä puhuttaessa käytetään usein myös sanoja kalvolevy, tietolevy, diskette, floppy disk ja lerppulevy. Kaikilla näillä tarkoitetaan kuitenkin samaa asiaa (suom. huom.).

Nyt on aika tallettaa ohjelma levykkeelle. Kirjoita sitä varten tietokoneelle tämä esimerkkiohjelma.

```
10 PRINT CHR$(12)
20 PRINT "SATUNNAISLUKUTAULUKKO"
30 PRINT
40 FOR N=1 TO 10
50  PRINT USING "# # #.# #" 100*RND
60 NEXT N
70 PRINT
```

Kun allaoleva komento annetaan, tallettuu ohjelma levykkeelle nimellä "SATLUKU".

```
SAVE SATLUKU
```

Kun talletus muutaman sekunnin kuluttua on päättynyt, kirjoittuu kuvaruudulle teksti ABC800.

Yksinkertaisimmassa tapauksessa ohjelman nimi muodostuu enintään kahdeksasta merkistä, joista ensimmäisen tulee olla kirjain. Ohjelman nimen pitäisi viitata ohjelman toimintaan.

Ohjelman noutaminen levykkeeltä

Tyhjennä tietokoneen sisäinen muisti komennolla NEW. Nouda sitten ohjelma SATLUKU-levykkeeltä tietokoneen sisäiseen muistiin komennolla:

```
LOAD SATLUKU
```

Tarkista komennolla LIST, että ohjelma todella noudettiin tietokoneeseen.

Ohjelma kasettinauhurilla

Jos tietokonejärjestelmässäsi on kasettinauhuri ainoana ulkoisena muistilaitteena, voit käyttää käskyjä SAVE ... ja LOAD ... vastaavalla tavalla kuin levykeyksikköä käytettäessä.

Ohjelmaa kasetille talletettaessa täytyy nauhurin olla asetettuna äänitystilaan. Tämän jälkeen anna esim. komento SAVE SATLUKU.

Kun ohjelma halutaan noutaa kasetilta tietokoneen sisäiseen muistiin, täytyy nauha ensin kelata alkuun ja nauhuri asettaa toistotilaan. Sitten annetaan esim. komento LOAD SATLUKU.

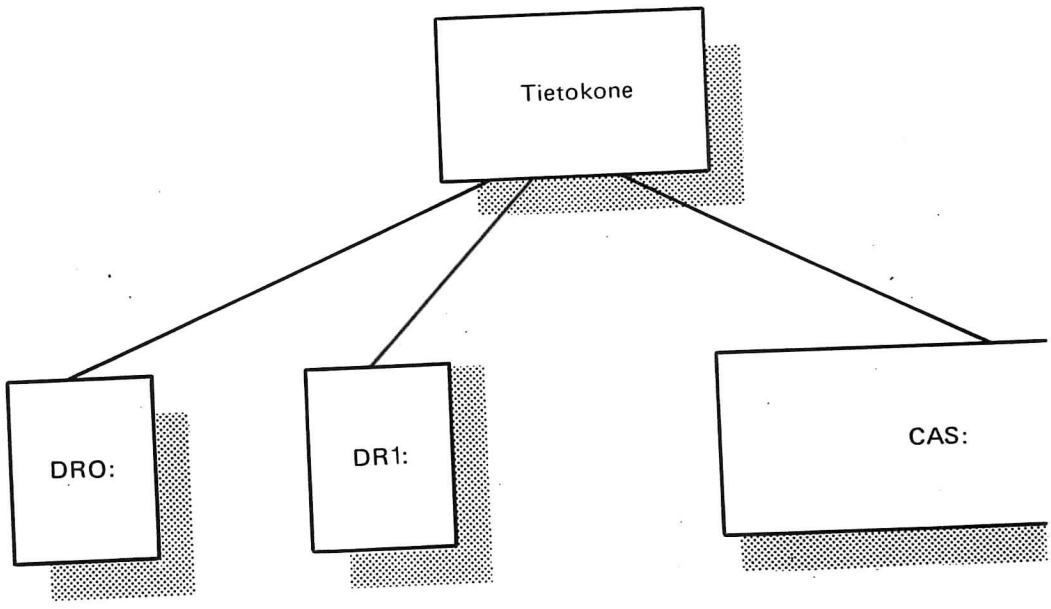
Useampia apumuisteja tietokonejärjestelmässä

Kun useampia tukimuistilaitteita on kytkettynä, täytyy tietenkin käskyissä ja komennoissa ilmoittaa, mitä yksikköä ne koskevat. Toisin sanoen yksikön ”osoite” täytyy ilmoittaa.

Levykeyksikkö sisältää yleensä kaksi erillistä levykeasemaa, joihin levykkeen voi sijoittaa. Käyttöohjeessa levymuistin asemesta käytetään nimityksiä DR0: (engl. DRive 0) ja vastaavasti DR1: (engl. DRive 1). Kun molemmat levykeasemat ja kasettimuisti ovat kytkettynä, voi tietokone siten viitata kolmeen eri tukimuistilaitteeseen, kumpaankin levyasemaan ja kasettimuistiin.

Jos yksikön nimeä ei anneta, valitsee tietokone ensisijaisesti levykeaseman 0 ja toissijaisesti levykeaseman 1. Muistiyksikön valintaa varten täytyy käskyjä ja komentoja edeltää DR0:, DR1: tai CAS:.

Tietokone asettaa siis yksiköt kuvion osoittamaan prioriteettijärjestykseen.



Prioriteetti 1

Yksikön nimi (DR1:) täytyy antaa joskus

Prioriteetti 2

Yksikön nimi (CA täytyy antaa aina

Levykkeen ohjelmakirjasto

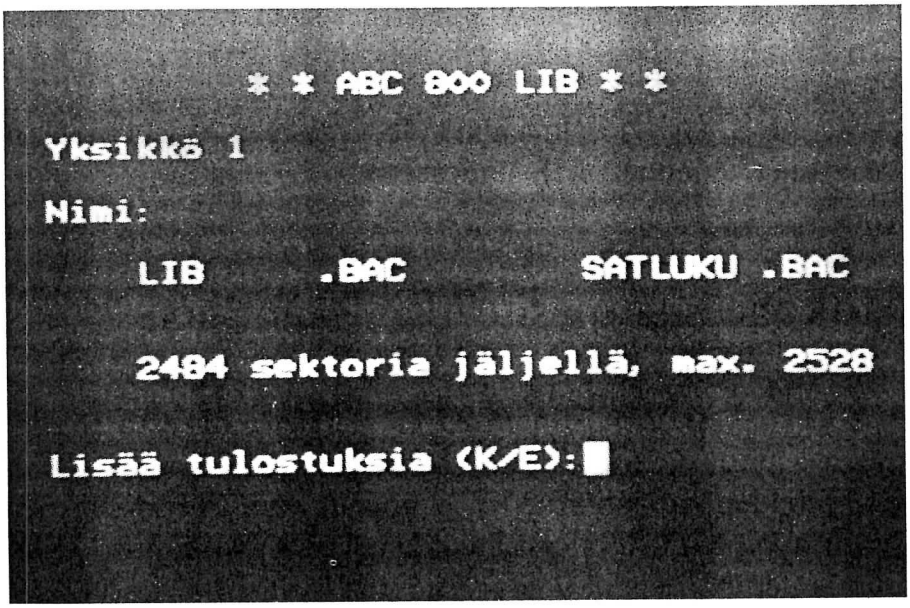
Levykkeelle talletetaan yleensä useita ohjelmia. Tietääkseen, mitä ohjelmia levyllä on ja mihin ne on levyllä talletettu, muodostaa tietokone levykkeelle ohjelmakirjaston.

Erityisellä BASIC-ohjelmalla, jota tavallisesti kutsutaan nimellä LIB (engl. LIBrary eli kirjasto), voi tietokone lukea koko kirjaston ja tulostaa sen kuvaruudulle tai kirjoittimelle. Levykkeillä, joille talletetaan ohjelmia, täytyy siksi olla talletettuna myös ohjelma LIB.

Jos ohjelma LIB on levykkeellä asemassa 0, voidaan ohjelma ajaa komennolla:

```
RUN LIB
```

Komento RUN LIB vastaa komentoa LOAD LIB, jota seuraa komento RUN. Toisin sanoen ohjelma noudetaan levykkeeltä ja sen suoritus alkaa.



Ohjelmanimeen SATLUKU on tietokone lisännyt liitteen .BAC. Tätä liitettä kutsutaan tiedostotyypiksi ja se osoittaa, että ohjelma on talletettu komennolla SAVE ... Komentoa LIST + ohjelmanimeä voi käyttää kuten komentoa SAVE ... tallettamaan ohjelmia levykkeille ja kaseteille. Ohjelmaan, joka on talletettu komennolla LIST ..., tietokone lisää tiedostotyypin .BAS.

Jos ohjelma LIB puuttuu levykkeeltä, antaa kone virheilmoituksen Error 21, kun komento RUN LIB annetaan. Tämä tarkoittaa, että tietokone ei löydä levykkeeltä ohjelmaa LIB. Yksi levyke, jolta LIB varmuudella löytyy, on levykeyksikön mukana seuraava järjestelmälevyke. Sijoita järjestelmälevyke oikeanpuoleiseen asemaan (DR1:) ja anna sitten uudestaan komento RUN LIB. Nyt näkyvät molempien levykkeiden sisällöt kuvaruudulla. Kun ohjelma LIB on nyt tietokoneen sisäisessä muistissa, voit samalla tallettaa sen tulevia tarpeita varten aseman 0 (DR0:) levykkeelle komennolla:

```
SAVE DRO:LIB
```

Poista järjestelmälevyke ja tarkista, että ohjelma LIB on nyt aseman 0 levykkeellä.

Ohjelman poistaminen levykkeeltä

Komennolla UNSAVE, jota seuraa ohjelman nimi, voi poistaa halutun ohjelman levykkeeltä. Poista ohjelma SATLUKU komennolla:

```
UNSAVE SATLUKU
```

Tarkistaaksesi, ettei ohjelma enää ole levykkeen ohjelmakirjastossa, voit ajaa ohjelman LIB uudestaan.

Tietokoneen kapasiteetin lisääminen

Tietokoneen keskusmuistin koko ei ole tietokoneessa toimivien ohjelmien ainoa kokoa rajoittava tekijä. Myös kytkettyjen apumuistilaitteiden varastointikyky ja nopeus vaikuttavat siihen, miten suuria ohjelmia koneessa voidaan ajaa. Apumuistilaitteelta vaaditaan, että ohjelmat voidaan noutaa hyvässä järjestyksessä ja nopeasti. Levykeyksikkö täyttää nämä vaatimukset, mutta kasettimuisti ei.

Laajojen ohjelmien laatimisessa menetellään siten, että ohjelma muodostetaan erillisistä osaohjelmista. Jokaisen osaohjelman loppuun liitetään käsky CHAIN "...", joka tarkoittaa, että seuraava ohjelmakappale ladataan apumuistista tietokoneen keskusmuistiin.

Tavallisesti kaikki muuttujat nollataan, kun käsky CHAIN "..." suoritetaan, mutta määrittelemällä eri osaohjelmien yhteiset muuttujat käskyllä COMMON ... voidaan näiden muuttujien nollautuminen välttää (katso HARJOITUSTEHTÄVIÄ).

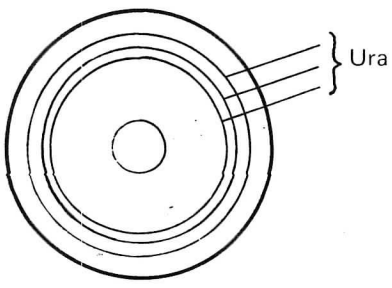
Opettele käyttämään järjestelmäohjelmia

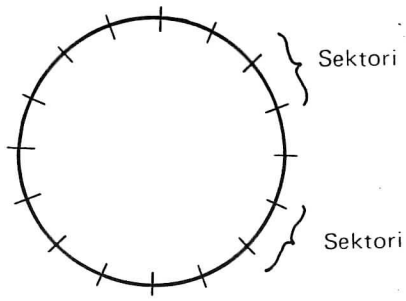
Olemme aikaisemmin näyttäneet, miten yksittäisiä ohjelmia talletetaan, noudetaan ja poistetaan levykkeeltä BASIC-komennoilla. Levykkeelle voi sopia satoja ohjelmia. Kun halutaan kopioida suuri määrä ohjelmia levykkeeltä toiselle, olisi epäkäytännöllistä tehdä se BASIC-komennoilla. Mm. tällaisen kopioinnin helpottamiseksi on tietokonejärjestelmässä useita erityisiä ohjelmia, niin sanottuja järjestelmäohjelmia.

Muita systeemiohjelmien toimintoja ovat mm. ohjelman poistaminen, levykkeiden nimeäminen ja uuden levykkeen alustaminen käyttöä varten. Tiettyjä järjestelmäohjelmia voi käyttää jopa antamaan tietokoneelle uusia ominaisuuksia, esim. virheilmoitusten antaminen selväkielisinä. Muutamia järjestelmäohjelmia on perusteellisesti kuvattu levykeyksikön käyttöohjeessa ja nämä ohjelmat ovat talletettuna levykkeelle nimeltä JÄRJESTELMÄLEVY. Annamme tässä pari esimerkkiä näiden ohjelmien käytöstä.

Uusien levykkeiden alustus

Kun ohjelma viedään tietokoneen keskusmuistista levykkeelle, tapahtuu tiedon talletus yhdelle tai useammalle uralle tekniikalla, joka muistuttaa nauhan äänittämistä. Jokainen levyke on tavallisesti jaettu 40 uraan. Jokainen ura on lisäksi jaettu useaan sektoriin, joita tavallisesti on 16 kappaletta kuten kuva osoittaa.





Jotta tietokone voisi löytää oikean uran ja sektorin, täytyy levykkeellä jo olla tietty informaatio ennen kuin sille voidaan tallettaa ohjelmia. Näytämme nyt miten käyttämättömän levykkeen alustus tapahtuu.

Sijoita järjestelmälevyke yksikköön 0 (DR0:) ja käyttämätön levyke yksikköön 1 (DR1:).

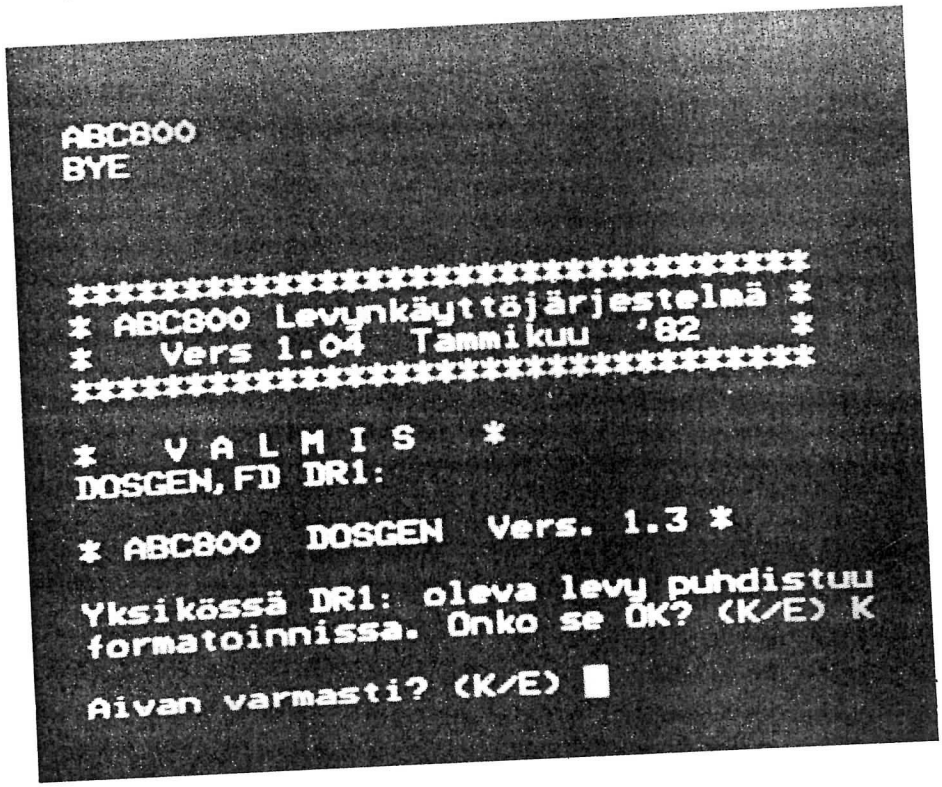
Voidaksesi käyttää järjestelmäohjelmia on sinun annettava komento:

```
BYE
```

Tämän jälkeen aja ohjelma, joka alustaa levykkeen yksikössä 1, komennolla:

```
DOSGEN,FD DR1:
```

Huomaa, että merkintöjen FD ja DR1: välissä täytyy olla välilyönti. Kun olet vastannut muutamiin tarkistuskysymyksiin, levykkeen alustus ja tarkastus alkaa.

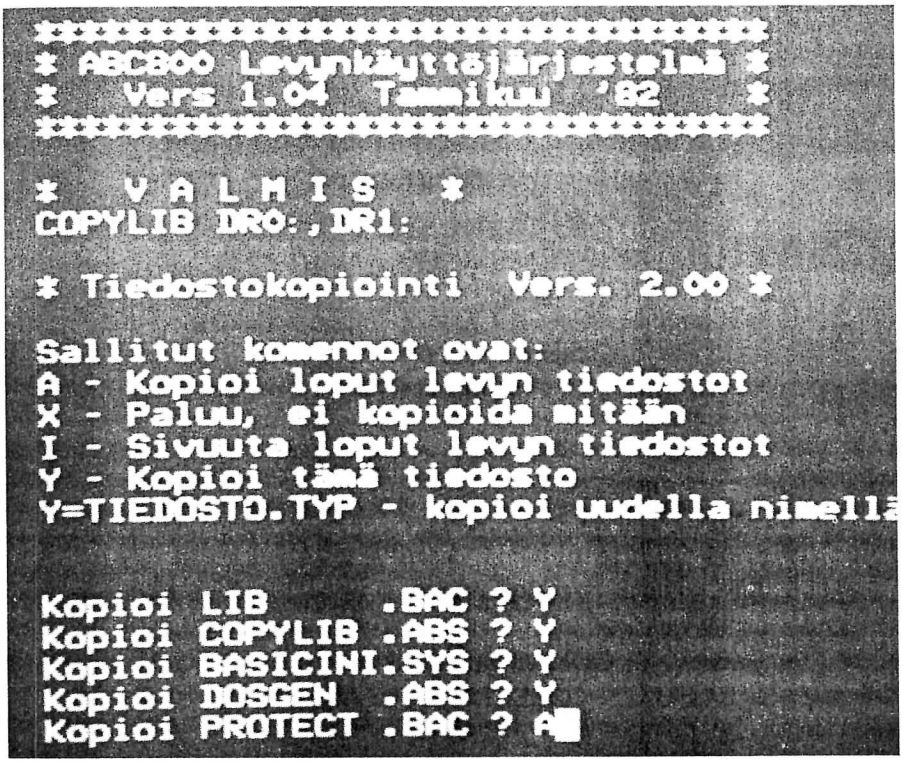


Muutaman minuutin kuluttua tietokone antaa ilmoituksen, että levyke on valmis käyttöön.

Levykkeiden kopioiminen

Usein levykkeelle ohjelman ja/tai datan muodossa talletettu tieto on käyttäjälleen hyvin arvokasta. Valitettavasti mitkään tallennuslaitteet eivät ole täysin varmoja. Luettaessa ja kirjoitettaessa levyke rasittuu mekaanisesti, mikä vähentää levykkeen ikää. Myös odottamaton sähkökatko ajon aikana saattaa aiheuttaa talletetun tiedon katoamisen. Lisäksi levyke voi helposti vaurioitua huolimattomassa käsittelyssä. Siksi onkin hyvin tärkeätä ottaa kopioita levykkeen sisällöstä. Voit esimerkiksi kopioida järjestelmälevykkeen sisällön levykkeelle, jonka juuri alustit käyttöä varten. Kopioiminen aloitetaan komennolla:

```
COPYLIB DRO:,DR1:
```



Nyt on mahdollista kopioida halutut ohjelmat levykkeeltä yksikössä 0 (DR0:) levykkeelle yksikössä 1 (DR1:). Kuvaruudulla näytetään yksikön 0 levykkeellä olevat ohjelmat. Vastaamalla ohjeiden mukaan voit valita, mitkä ohjelmat kopioidaan. Esimerkiksi vastaus A (All), jota seuraa RETURN-näppäimen painaminen, tarkoittaa, että loput ohjelmat kopioidaan. Pelkkä RETURN-näppäimen painaminen tarkoittaa, että kyseistä ohjelmaa ei kopioida. Kun kaikki ohjelmat on esitetty kuvaruudulla, alkaa haluttujen ohjelmien kopioiminen. Muutaman minuutin kuluttua kopioiminen päättyy.

Jos haluat lopettaa järjestelmäohjelmien käytön ja palata BASIC-tulkin alaisuuteen, anna komento:

```
$BAS
```

Yksikössä 1 sijaitseva levyke sisältää nyt muutamia järjestelmäohjelmia. On käytännöllistä varustaa ohjelmien talletukseen tarkoitetut levykkeet näillä järjestelmäohjelmilla jo alustettaessa. Kaikkia levykkeitä voi tällöin käyttää esim. levykkeiden kopioimiseen ja alustamiseen.

Yhteenveto

Käskyt

CHAIN "... ... ..."
Siirrä annettu ohjelma kasetilta tai levykkeeltä ABC800:aan ja aloita ohjelman ajo. Käskyllä on sama vaikutus kuin käskyllä RUN ... ...
Katso myös COMMON...

```
10 CHAIN "CAS: LUETTELO.007"
60 CHAIN "OHJELMA2"
90 CHAIN A$
```

COMMON ...
Annettuja muuttujia ei nollata, kun käsky CHAIN "... ... ..." suoritetaan. Annettavat merkkijonomuuttujat on dimensioitava. COMMON-ilmoituksia sisältävät ohjelmarivit on sijoitettava ohjelman alkuun.

```
10 COMMON X, Y, Z
20 COMMON X$(30) = 100, Y$(20) = 200
```

Komennot

$BAS
Aiheuttaa paluun BASIC:iin. Vastaa RESET-näppäimen käyttöä. Vrt. BYE.

BYE
Lopettaa BASIC-tulkin käytön. Sulkee kaikki tiedostot.

LIST ... ...
Siirtää ABC800:n muistissa olevan ohjelman tiedostoksi kasetille tai levykkeelle kääntämättömässä muodossa (tekstimuodossa). Jos tiedostotyyppi jätetään pois, käytetään tyyppiä .BAS.

```
LIST OSAOHJE,20—40
LIST DR1: SORTTAUS
LIST CAS: TESTI.001
```

LOAD ... ...
Siirtää ilmoitetun ohjelman kasetilta tai levykkeeltä ABC800:n muistiin. Poistaa aikaisemman ohjelman muistista.

```
LOAD BUDJETTI
LOAD DR1: KALKYYLI.001
LOAD CAS:    (Siirrä ensimmäinen kasetilta
              löytyvä ohjelma.)
```

RUN ... ...
Vastaa komentoa LOAD ... ..., jota seuraa komento RUN. Vrt. LOAD ... ...
Yksikön nimi sekä tiedostotyypit .BAC ja .BAS voidaan jättää pois.

```
RUN CAS: LUETTELO.007
```

SAVE ... ...
Siirtää ohjelman ABC800:n muistista tiedostoksi kasetille tai levykkeelle koneen sisäisen koodin muodossa. Jos tiedostotyyppi jätetään ilmoittamatta, käytetään tyyppiä .BAC.
Vrt UNSAVE ... ...

```
SAVE SORTTAUS
SAVE DR1: SORTTAUS
SAVE CAS: TESTI.001
```

UNSAVE ... ...
Poistaa ilmoitetun tiedoston levyltä. Jos tiedostotyyppi jätetään pois, käytetään ensisijaisesti tyyppiä .BAC ja toissijaisesti tyyppiä .BAS.
Vrt SAVE ... ...

Harjoitustehtäviä

* 6.1 Seuraavat kolme ohjelmaa osoittavat, miten käskyä CHAIN "... ... ..." voidaan käyttää säätelemään ohjelmien levyltä noutamista ja suoritusta. Talleta ohjelmat levykkeelle nimillä EX1, EX2 ja EX. Käynnistä sitten ohjelma EX1, ja tarkkaile missä järjestyksessä ohjelmat suoritetaan.

```
EX1 10 PRINT CHR$(12)
    20 PRINT "OSTA MAUKKAITA KEITETTYJÄ NAKKEJA"
    30 PRINT
    40 PRINT "NIILON NAKKIBAARI"
    50 FOR N=1 TO 5000 : NEXT N
    60 CHAIN "EX2"
```

```
EX2 10 PRINT CHR$(12)
    20 PRINT "HYVIÄ PITKOJA PAISTAA VAIN"
    30 PRINT
    40 PRINT "PAAKARI PULLA"
    50 FOR N=1 TO 5000 : NEXT N
    60 CHAIN "EX3"
```

```
EX3 10 PRINT CHR$(12)
    20 PRINT "ME NÄYTÄMME MAAILMAN"
    30 PRINT "HUONOIMPIA VIDEOFILMEJÄ"
    40 PRINT
    50 PRINT "VILLEN VIDEOKELLARI"
    60 FOR N=1 TO 5000 : NEXT N
    70 CHAIN "EX1"
```

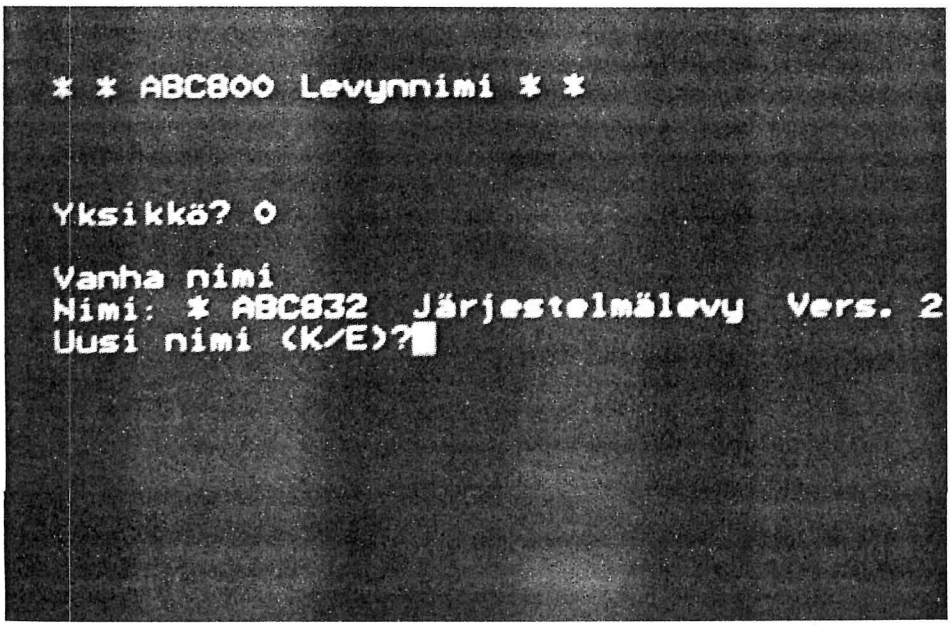
6.2 Jos kaikkien muuttujien ei haluta nollautuvan CHAIN-käskyn suorituksessa, ilmoitetaan muuttujat, joiden halutaan säilyvän muuttumattomina COMMON-lauseessa. Seuraavassa esimerkissä muuttuja A$ säilyttää arvonsa, kun seuraava ohjelma suoritetaan. Huomaa, että kaikki COMMON-lauseessa esiintyvät merkkijonomuuttujat on dimensioitava. Talleta ohjelmat levykkeelle SAVE-komennolla ja nimillä SYÖTTÖ ja TULOSTUS, ja aja sitten ohjelma SYÖTTÖ.

```
10 COMMON A$ = 100
20 PRINT
30 PRINT " *** OHJELMA: SYÖTTÖ ***"
40 PRINT
50 INPUT "SYÖTÄ NIMI / (LOPPU): "A$
60 PRINT
70 IF A$ = "LOPPU" THEN GOTO 80 ELSE CHAIN "TULOSTUS"
80 PRINT "AJO PÄÄTTYNYT"
```

```
10 COMMON A-ö- = 100
20 PRINT CHR-ö-(12)
30 PRINT "*** OHJELMA: TULOSTUS ***"
40 PRINT
50 INPUT "EDELLINEN SYÖTTÖ: "A-ö-
60 PRINT
70 CHAIN "SYÖTTÖ"
```

6.3 Sen lisäksi, että levykkeitä voi merkitä liimaamalla niihin etiketin, voi tietoja levykkeen sisällöstä tallettaa suoraan levykkeelle. Tämä tapahtuu käyttämällä apuna järjestelmäohjelmaa NAMEDISK.BAC.

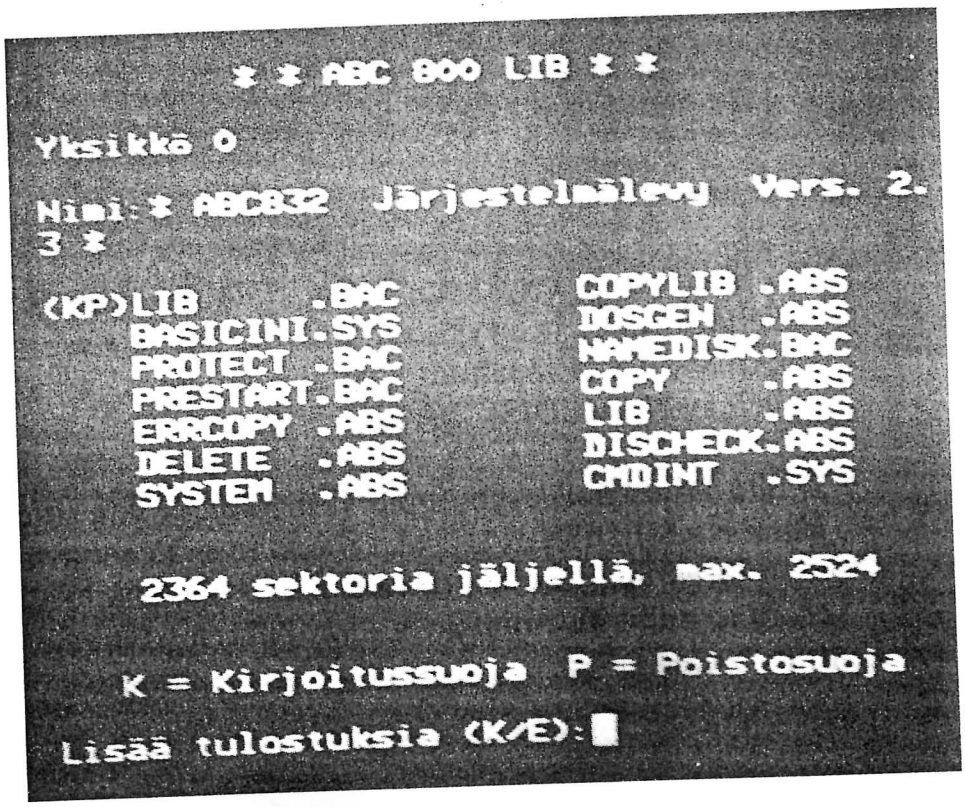
Sijoita järjestelmälevyn kopio yksikköön 0. Aja sitten ohjelma NAMEDISK.BAC. Merkitsemiseen käytettävä teksti saa olla pituudeltaan korkeintaan 120 merkkiä. Valokuva osoittaa miltä kuvaruutu voi ajon jälkeen näyttää.



6.4 Välttääksesi levykkeelle talletettujen ohjelmien tuhoutumisen voit liimata teipin levykkeen suojauran päälle (katso kuvaa sivulla 90).

Vaihtoehtoisesti voi yksittäisiä ohjelmia kirjoitus- ja/tai poistosuojata käyttäen apuna järjestelmäohjelmaa PROTECT.BAC. Aja tämä ohjelma, ja varusta ohjelma LIB.BAC kirjoitussuojalla.

Aja sitten ohjelma LIB käyttäen vaihtoehtoa 3, eli tiedostostatus, nähdäksesi, että ohjelma on todella kirjoitussuojattu kuten kuva osoittaa.



7

Ohjelman rakenne

Kommentit ohjelmassa

Ohjelmien käsittelyn helpottamiseksi on käytännöllistä liittää kaikkiin ohjelmiin samantapainen aloitusosa. Tämä aloitusosa voisi sisältää mm. ohjelman nimen, ohjelmoijan ja päivämäärän. Ohjelman alku voi näyttää vaikka tältä:

```
100 REM OHJELMA      : KILPA1
110 REM OHJELMOIJA   : MAIJA SUOMINEN
120 REM PÄIVÄYS      :1982-05-15
130 REM TIETOKONE    : ABC 800
```

Käskyä REM ... (engl. REMark) käytetään erottamaan kommentit ohjelmatekstistä. REM-lause ei aiheuta mitään toimintaa ohjelman suorituksessa.

Muuttujien dimensiointi

Seuraavaa ohjelmakappaletta, jonka aloitusosa on riveillä 100—130, voidaan käyttää kilpailun 15 osanottajan nimen ja lähtönumeron rekisteröimiseen. Osanottajat voidaan syöttää missä järjestyksessä hyvänsä.

```
140 DIM K$(15)=40
150 WHILE V$<>"LOPPU"
160  REM ------- TULOSTUS ----------
170  PRINT CHR$(12)
180  PRINT "FINLANDIA-HIIHTO"
190  PRINT
200  PRINT "LÄHTÖNUMERO", "NIMI"
210  PRINT
220  FOR N=1 TO 15
230    PRINT N,K$(N)
240  NEXT N
250  PRINT
```

```
260  REM ------- SYÖTTÖ ----------
270  INPUT "MIKÄ LÄHTÖNUMERO?              "N
280  IF N>15 THEN N=15
290  INPUT "OSANOTTAJAN NIMI/(LOPPU)?"V$
300  K$(N)=V$
310 WEND
320 REM ------- LOPETUS ----------
330 PRINT "OHJELMAN AJO PÄÄTTYNYT"
340 END
```

Kun ohjelmarivi 140 suoritetaan, varataan tilaa (DIMensioidaan) 16 merkkijonomuuttujalle, D$(0):sta D$(15):een. Samalla rajoitetaan jonomuuttujien määrä korkeintaan 40 merkkiin. Jos merkkijonon suurinta pituutta ei ilmoiteta DIM-lauseessa, asetetaan maksimipituudeksi automaattisesti 80 merkkiä. Huomaa, että kaikki taulukkomuuttujat, joissa indeksi saa suuremman arvon kuin 10, täytyy dimensioida (katso yhteenvetoa).

Koska DIM-lauseen täytyy esiintyä ohjelmassa ennen vastaavan taulukkomuuttujan käyttöä, on kätevintä sijoittaa DIM-lauseet ohjelman alkuun.

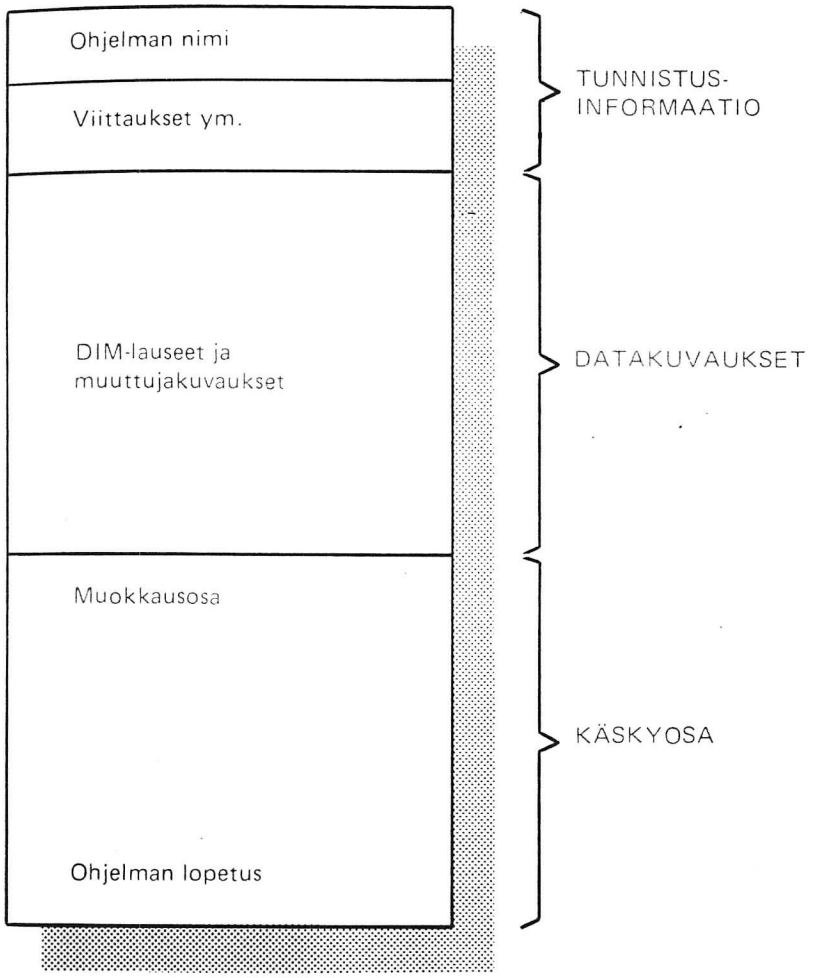
Riveillä 160, 260 ja 320 on REM-lauseita käytetty erottamaan ohjelman kolmea erillistä kappaletta. Kappaleita TULOSTUS ja SYÖTTÖ suoritetaan kunnes muuttuja V$ saa arvon "LOPPU".

Kappale LOPETUS sisältää rivillä 340 käskyn END, joka päättää ohjelman suorituksen. END-käskyä käytetään osoittamaan ohjelman loogista loppua.

Kuten aikaisemmista esimerkeistä on käynyt selville, voidaan END-käsky jättää pois, jos ohjelman suoritus päättyy sen viimeiselle riville. Kuitenkin on hyvä tapa merkitä ohjelman loogista loppua END-käskyllä, sillä aina ohjelman looginen loppu ei ole viimeisellä ohjelmarivillä.

Ohjelman rakenne

Tähän mennessä käsitellyt asiat voidaan esittää yhteenvetona seuraavassa kuviossa:



Kun käytät ehdotettua mallia ohjelmia suunnitellessasi, tulee ohjelmistasi selkeitä ja siten helpompia ymmärtää. Myös ohjelmaan myöhemmin tehtävien muutosten teko helpottuu, kun ohjelma on järkevästi rakennettu.

Ohjelman ajon keskeytyminen

Vaikka ohjelma KILPA1 onkin muodollisesti täysin kunnollinen, voi virheellinen syöttö johtaa odottamattomaan keskeytykseen ohjelman suorituksessa. Jos ohjelmaa ajettaessa annat lähtönumeron kirjaimilla (esim. KOLME), keskeytyy ohjelman ajo ja seuraava ilmoitus tulostuu kuvaruudulle:

```
ERROR 210 IN LINE 270.
```

Virhe 210 (katso ohjelmointikorttia) rivillä 270 tarkoittaa, että lukumuuttujalle N annettiin arvo (esim. KOLME), jota ei kirjoitettu numeroilla. Koska lukumuuttujat voivat saada vain numeroarvoja, on tietokone ratkeamattoman ongelman edessä ja siksi ohjelman ajo täytyy keskeyttää.

Jos levyke, jolla on järjestelmäohjelma BASICINI.SYS, on koneen levy-yksikössä, tulostuu virheilmoitus selväkielisenä.

Keskeytynyttä ajoa voidaan jatkaa riviltä, jolla virhe tapahtui, komennolla:

```
CONTINUE
```

Ennen keskeytystä asetetut muuttujat säilyttävät arvonsa, kun ohjelman ajoa jatketaan CONTINUE-komennolla.

Myös kun ohjelman suoritus on keskeytetty painamalla yhtaikaa kontrollinäppäintä CTRL ja näppäintä C , voidaan suoritusta jatkaa samalla tavalla komennolla CONTINUE.

Jos tarkoituksellisesti halutaan keskeyttää ohjelman suoritus tiettyyn kohtaan siten, että siitä voi jatkaa CONTINUE-komennolla, käytetään käskyä STOP (katso HARJOITUSTEHTÄVIÄ).

Virheiden käsittely ohjelmassa

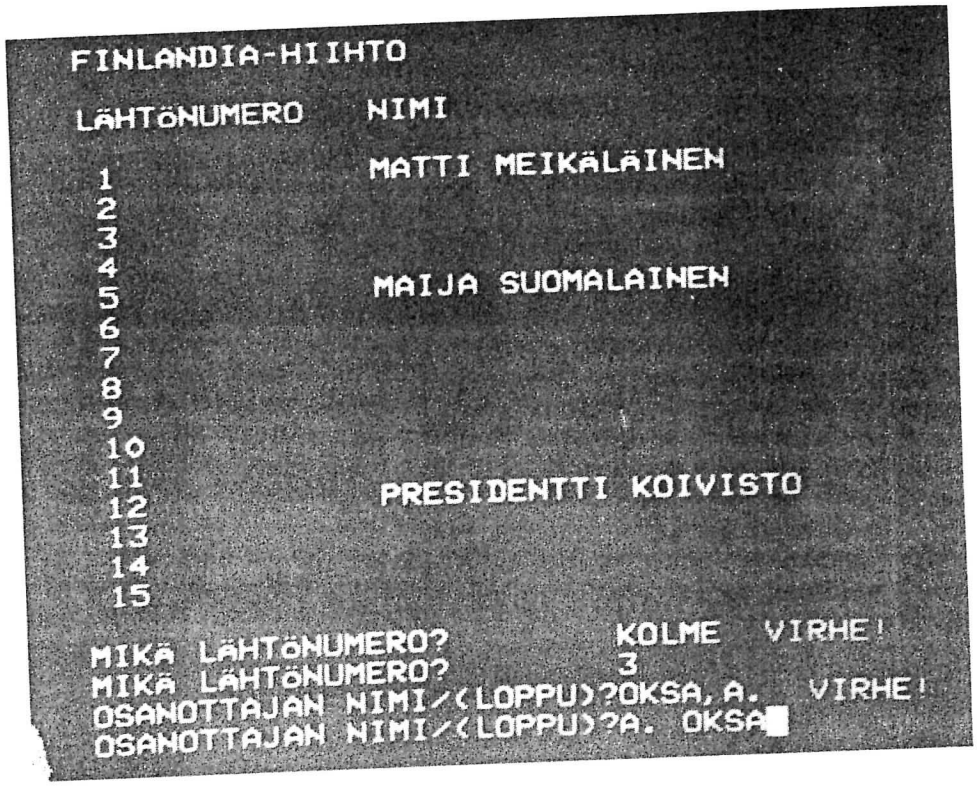
Helppokäyttöisen ohjelman suoritus ei tietenkään saa keskeytyä virheelliseen syöttöön. Siksi on olemassa käsky ON ERROR GOTO ..., joka estää virhettä johtamasta ohjelman keskeytymi-

seen. Tämä käsky tarkoittaa, että virheen sattuessa ohjelman suoritusta jatketaan toisesta paikasta. Lisää ohjelmaan allaolevat rivit 265 ja 350—370.

```
100 REM OHJELMA       : KILPA2
110 REM OHJELMOIJA    : MAIJA SUOMINEN
120 REM PÄIVÄYS       : 1982-05-15
130 REM TIETOKONE     : ABC 800
140 DIM K$(15)=40
150 WHILE V$<>"LOPPU"
160  REM ------- TULOSTUS ----------
170  PRINT CHR$(12)
180  PRINT "FINLANDIA-HIIHTO"
190  PRINT
200  PRINT "LÄHTÖNUMERO", "NIMI"
210  PRINT
220  FOR N=1 TO 15
230    PRINT N,K$(N)
240  NEXT N
250  PRINT
260  REM ------- SYÖTTÖ ----------
265  ON ERROR GOTO 350
270  INPUT "MIKÄ LÄHTÖNUMERO?          "N
280  IF N>15 THEN N=15
290  INPUT "OSANOTTAJAN NIMI/(LOPPU)?"V$
300  K$(N)=V$
310 WEND
320 REM ------- LOPETUS ----------
330 PRINT "OHJELMAN AJO PÄÄTTYNYT"
340 END
350 REM ------- VIRHEEN KÄSITTELIJÄ ----
360 PRINT "          VIRHE!"
370 RESUME
```

Jos rivin 265 suorituksen jälkeen tapahtuu virhe, jatketaan ohjelman suoritusta riviltä 350. Sopiva virheilmoitus tulostetaan sitten, kun rivi 360 suoritetaan. Käsky RESUME aiheuttaa suorituksen jatkumisen sen rivin alusta, jolla virhe tapahtui eli riviltä 270. Ajaessasi ohjelman huomaat, että N saa nyt kelvollisen arvon rivillä 270.

Mitä tapahtuu, jos virhe syntyy muuttujan S$ syötössä rivillä 290? Myös tässä tapauksessa suoritetaan ohjelmakappale VIRHEENKÄSITTELIJÄ, jotta rivillä 290 tapahtuisi kelvollinen syöttö.



Käytä kuvaavia muuttujien nimiä

Tähän mennessä olemme käyttäneet muuttujien niminä lyhenteitä. Edellisissä esimerkeissä käytimme esim. muuttujaa N kuvaamaan numeroa. Muuttujien merkityksen ymmärtäminen helpottuu, kun muuttujien nimet valitaan näin:

N	korvataan muuttujalla	Numero
V$	korvataan muuttujalla	Vastaus$
K$(N)	korvataan muuttujalla	Kilpailija$(Numero)

Kun käsky EXTEND sijoitetaan ohjelman alkuun, hyväksyy tietokone mielivaltaisen pitkiä muuttujan nimiä. Jotta tietokone voisi erottaa muuttujien nimet ja BASIC-käskyt toisistaan, täytyy sanat ja ilmaukset erottaa välilyönneillä.

Jos ohjelmassa KILPA2 käytämme ''itseselittäviä'' muuttujanimiä, tulee ohjelmasta seuraavan näköinen:

```
100 REM OHJELMA        : KILPA3
110 REM OHJELMOIJA     : MAIJA SUOMINEN
120 REM PÄIVÄYS        : 1982-05-15
130 REM TIETOKONE      : ABC 800
135 EXTEND
140 DIM Kilpailija$(15)=40
150 WHILE Vastaus$<>"LOPPU"
160  REM ------- TULOSTUS ----------
170  PRINT CHR$(12)
180  PRINT "FINLANDIA-HIIHTO"
190  PRINT
200  PRINT "LÄHTÖNUMERO", "NIMI"
210  PRINT
220  FOR Numero=1 TO 15
230    PRINT Numero,Kilpailija$(Numero)
240  NEXT Numero
250  PRINT
260  REM ------- SYÖTTÖ ----------
265  ON ERROR GOTO 350
270  INPUT "MIKÄ LÄHTÖNUMERO?              "Numero
280  IF NUMERO>15 THEN Numero=15
290  INPUT "OSANOTTAJAN NIMI/(LOPPU)?"Vastaus$
300  Kilpailija$(Numero)=Vastaus$
310 WEND
320 REM ------- LOPETUS ----------
330 PRINT "OHJELMAN AJO PÄÄTTYNYT"
340 END
350 REM ------- VIRHEEN KÄSITTELIJÄ ----
360 PRINT "        VIRHE!"
370 RESUME
```

Jos tietokone antaa virheilmoituksen Error 220, ''Ei ymmärrä'', on syynä luultavasti se, että et ole käyttänyt välilyöntejä oikein.

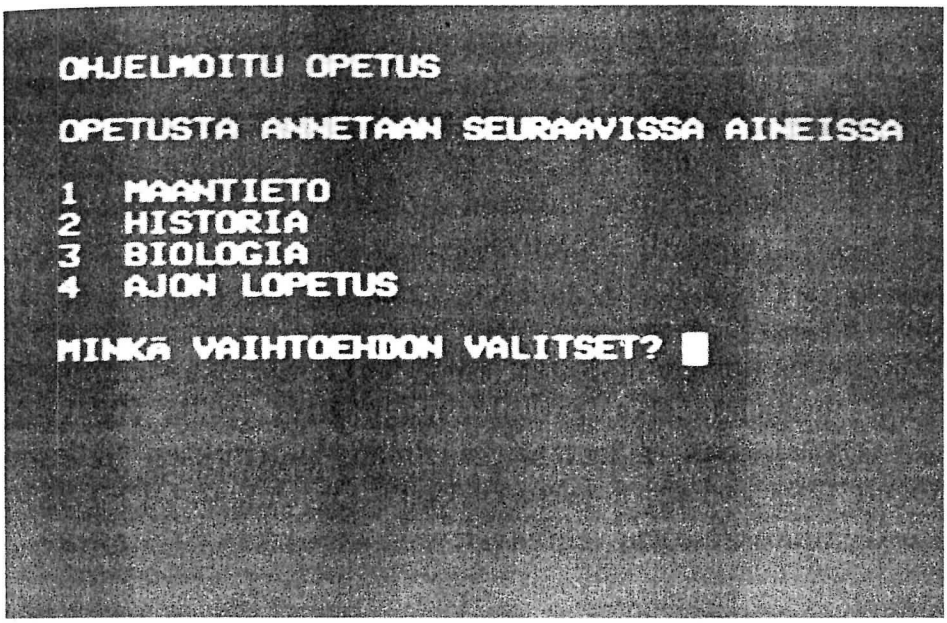
Mahdollisuus käyttää mielivaltaisen pitkiä muuttujanimiä poistetaan käskyllä NO EXTEND.

Menu-tekniikka

Usein tietokoneohjelmalla voi suorittaa useita erilaisia tehtäviä. Useimmissa tapauksissa syötettävä data vastaa esitettyä alkutekstiä. Jos alkuteksti sisältää useita vaihtoehtoja, täytyy alkuteksti muotoilla menuksi, toisin sanoen kaikkien vaihtoehtojen listaksi. Seuraava ohjelmaesimerkki näyttää, miten menu voidaan muotoilla.

```
100 REM OHJELMA       : OPETUS1
110 REM OHJELMOIJA    : MAIJA SUOMINEN
120 REM PÄIVÄYS       : 1982-05-20
130 REM TIETOKONE     : ABC 800
140 EXTEND
150 PRINT CHR$(12)
160 PRINT "OHJELMOITU OPETUS"
170 PRINT
180 PRINT "OPETUSTA ANNETAAN SEURAAVISSA AINEISSA"
190 PRINT
200 PRINT "1 MAANTIETO"
210 PRINT "2 HISTORIA"
220 PRINT "3 BIOLOGIA"
230 PRINT "4 AJON LOPETUS"
240 PRINT
250 ON ERROR GOTO 420
260 INPUT "MINKÄ VAIHTOEHDON VALITSET ? "Vaihtoehto
270 PRINT
280 ON Vaihtoehto GOTO 300,340,380,450
290 GOTO 150
300 REM ------- VAIHTOEHTO 1 ----------
310 PRINT "HALTIATUNTURI ON SUOMEN KORKEIN TUNTURI"
320 FOR Kierros=1 TO 3000 : NEXT Kierros
330 GOTO 290
340 REM ------- VAIHTOEHTO 2 ----------
350 PRINT "30-VUOTINEN SOTA OLI VUOSINA 1618—1648"
360 FOR Kierros=1 TO 3000 : NEXT Kierros
370 GOTO 290
380 REM ------- VAIHTOEHTO 3 ----------
390 PRINT "KOTIKÄRPÄNEN ON MAAILMAN VAARALLISIN ELÄIN"
400 FOR Kierros=1 TO 3000 : NEXT Kierros
410 GOTO 290
420 REM ------- VIRHEEN KÄSITTELIJÄ ----
430 PRINT " VIRHE !"
440 RESUME 260
450 REM ------- LOPETUS ----------
460 PRINT "OHJELMAN AJO PÄÄTTYNYT"
470 END
```

Menu tulostetaan riveillä 150—270.



Vastauksena oleva luku 1, 2, 3 tai 4 sijoitetaan lukumuuttujaan Vaihtoehto. Kun rivi 280 suoritetaan, jatkuu ohjelman suoritus eri riveiltä riippuen muuttujan Vaihtoehto arvoista. Periaate käy ilmi taulukosta:

Vaihtoehto = 1	tarkoittaa: hyppää riville 300
Vaihtoehto = 2	tarkoittaa: hyppää riville 340
Vaihtoehto = 3	tarkoittaa: hyppää riville 380
Vaihtoehto = 4	tarkoittaa: hyppää riville 450

Mitä tapahtuu, jos muuttujalle Vaihtoehto annetaan vahingossa arvo 5? Koska rivillä 280 on vain neljä vaihtoehtoista rivinumeroa, aiheutuu virhe, joka tavallisesti keskeyttäisi ohjelman suorituksen. Koska rivi 250 on tätä ennen suoritettu, siirrytäänkin sen sijaan rutiiniin VIRHEENKÄSITTELIJÄ.

Ohjelman suoritusta ei kuitenkaan voida jatkaa ennen kuin muuttuja Vaihtoehto on saanut kelvollisen arvon ja siksi ohjelmassa siirrytään riville 260, kun käsky RESUME 260 suoritetaan.

Aliohjelmien käyttö ohjelmoinnissa

Kun yhdessä ohjelmassa esiintyy useita samanlaisia ohjelmakappaleita, voidaan nämä korvata aliohjelmilla. Ohjelmassamme OPETUS1 on samanlainen viivytyssilmukka kolmessa eri kohdassa, nimittäin riveillä 320, 360 ja 400. Nämä ohjelmakappaleet voidaan korvata ohjelman loppuun sijoitettavalla aliohjelmalla. Jotta aliohjelma suoritettaisiin oikeissa kohdissa, sijoitetaan aliohjelmakutsut riveille 320, 360 ja 400.

Aliohjelmakutsu muodostuu käskystä GOSUB ja rivinumerosta, jolta aliohjelma alkaa. Aliohjelma päättyy rivillä 500 käskyyn RETURN, joka aiheuttaa ohjelman suorituksen jatkumisen GOSUB-käskyä seuraavasta käskystä. Ohjelmaan OPETUS1 tehtävät muutokset ovat seuraavat:

```
100 REM OHJELMANIMI: OPETUS2
320 GOSUB 480
360 GOSUB 480
400 GOSUB 480
480 REM -------- ALIOHJELMA --------
490 FOR Kierros=1 TO 3000 : NEXT Kierros
500 RETURN
```

Järjestelmällisesti aliohjelmia käyttämällä tulee ohjelmista sekä lyhyempiä että selkeämpiä.

Funktiot korvaavat aliohjelmat

Käyttämällä funktioita aliohjelmien sijasta ohjelmiin saadaan vielä parempi rakenne. Tämä johtuu siitä, että funktion kutsussa tarvitsee antaa ainoastaan funktion nimi. Kun siis kutsutaan funktiota, ei tarvitse antaa rivinumeroa.

Ohjelmakappale, joka määrittelee funktion, aloitetaan käskyllä DEF FN ... ja lopetetaan käskyyn FNEND. Ohjelmakappaleessa täytyy lisäksi esiintyä käsky RETURN ... Funktio, joka aiheuttaa noin kolmen sekunnin viivytyksen, muotoillaan seuraavasti:

```
490 DEF FNViivytys
500  FOR Kierros=1 TO 3000 : NEXT Kierros
510  RETURN Eimitään
520 FNEND
```

Rivillä 490 annetaan funktion nimi Viivytys. Funktion tarkoitus on aiheuttaa viivytys ja tämä tapahtuu rivillä 500. Funktion täytyy aina palauttaa jokin arvo ja siksi ''palautetaan'' muuttujan Eimitään arvo rivillä 510.

Tässä tapauksessa funktion tehtävänä oli aiheuttaa viivytys jonkin funktion arvon laskemisen sijasta. Palautettava funktion arvo eli muuttujan Eimitään arvo on siis merkityksetön, mutta RETURN-lauseen täytyy esiintyä funktion määrittelyssä.

Funktioita kuten aliohjelmiakin voi käyttää monella eri tavalla. Lisää esimerkkejä on HARJOITUSTEHTÄVISSÄ.

Versiossa OPETUS3 on menun vaihtoehdot 1, 2 ja 3 toteutettu aliohjelmina. Rivillä 280 on siksi käsky ON ... GOSUB ... (vertaa ON ... GOTO ...), joka aiheuttaa eri aliohjelmakutsun muuttujan Vaihtoehto arvon mukaan.

Jokaisessa aliohjelmassa käytetään funktiota Viivytys. Funktion kutsu tapahtuu antamalla muuttujalle Odota funktion Viivytys arvo. Tämä tapahtuu riveillä 350, 390 ja 430.

```
100 REM OHJELMA       : OPETUS3
110 REM OHJELMOIJA    : MAIJA SUOMINEN
120 REM PÄIVÄYS       : 1982-05-20
130 REM TIETOKONE     : ABC 800
140 EXTEND
150 PRINT CHR$(12)
160 PRINT "OHJELMOITU OPETUS"
170 PRINT
180 PRINT "OPETUSTA ANNETAAN SEURAAVISSA AINEISSA"
190 PRINT
200 PRINT "1 MAANTIETO"
210 PRINT "2 HISTORIA"
220 PRINT "3 BIOLOGIA"
230 PRINT "4 AJON LOPETUS"
240 PRINT
250 ON ERROR GOTO 450
260 INPUT "MINKÄ VAIHTOEHDON VALITSET ? "Vaihtoehto
270 PRINT
280 ON Vaihtoehto GOSUB 330,370,410,300
290 GOTO 150
300 REM ------- LOPETUS ----------
310 PRINT "OHJELMAN AJO PÄÄTTYNYT"
320 END
330 REM ------- VAIHTOEHTO 1 ----------
340 PRINT "HALTIATUNTURI ON SUOMEN KORKEIN TUNTURI"
350 LET Odota=FNViive
360 RETURN
370 REM ------- VAIHTOEHTO 2 ----------
380 PRINT "30-VUOTINEN SOTA OLI 1618—1648"
390 LET Odota=FNViive
400 RETURN
410 REM ------- VAIHTOEHTO 3 ----------
420 PRINT "KÄRPÄNEN ON MAAILMAN VAARALLISIN ELÄIN"
430 LET Odota=FNViive
440 RETURN
450 REM ------- VIRHEEN KÄSITTELIJÄ ----
460 PRINT " VIRHE !"
470 RESUME 260
480 REM ------- VIIVE.DEF ----------
490 DEF FNViive
500   FOR Kierros=1 TO 3000 : NEXT Kierros
510   RETURN Eimitään
520 FNEND
```

Ohjelman looginen loppu on ohjelmarivillä 320 END.

Selkeyden vuoksi aliohjelmat, funktiot ja virherutiini on sijoitettu ohjelman loppuun.

Ohjelman osien talletus levykkeelle

Monia yhtä ohjelmaa varten tehtyjä ohjelmakappaleita voidaan käyttää myös toisissa ohjelmissa. Ohjelmakappale VIIVE.DEF, joka määritteli funktion Viivytys riveillä 490—520 ohjelmassa OPETUS3, voi olla käytännöllinen myös muissa yhteyksissä.

Komennolla ERASE ... voidaan poistaa se osa ohjelmaa, jota ei haluta tallettaa.

```
ERASE 100—470
```

Komennolla REN ... annetaan kappaleelle suuret rivinumerot myöhemmän käytön helpottamiseksi. Sallitut rivinumerot ovat 1-65535.

```
REN 10000
```

Komennolla LIST ... talletetaan ohjelmakappale levykkeelle.

```
LIST VIIVE.DEF
```

Tiedostotyyppiä .DEF käytetään tässä ilmoittamaan, että levykkeelle talletetaan funktion määritelmä eikä valmista ohjelmaa. Vastaavalla tavalla voidaan talletettuihin aliohjelmiin liittää tiedostotyyppi .SUB.

Jos levykkeeltä noudetaan ohjelma komennolla LOAD ..., poistuu aikaisemmin koneessa ollut ohjelma keskusmuistista. Kun ohjelmaan halutaan liittää levykkeelle talletettu kappale, täytyy käyttää komentoa MERGE ... Huomaa, että MERGE-komennolla noudettavan ohjelman täytyy olla talletettu levykkeelle komennolla LIST ...

Yhteenveto

Käskyt

DEF FN ... (...) LOCAL ...
Määrittelee oman monirivisen funktion. Ilmoituksella LOCAL ... määritellään paikalliset muuttujat. LOCAL ... voidaan jättää pois. Funktion arvo palautetaan käskyllä RETURN ... Käsky FNEND lopettaa määrittelyn.

```
100 DF FNY(R) LOCAL S
200  S = R/180
300  RETURN S * R
400 FNEND
```

DIM ...
Varaa tilaa kenttämuuttujille. Pienin indeksi on normaalisti 0, mutta se voidaan muuttaa ykköseksi käskyllä OPTION BASE 1.

```
10 DIM P(20)
```

Varaa tilaa 21 liukulukumuuttujalle P(0) — P(20).

```
20 DIM P$ = 1000
```

Varaa tilaa merkkijonolle, jossa on enintään 1000 merkkiä.

```
30 DIM P$(20) = 10
```

Varaa tilaa 21 merkkijonomuuttujalle, jossa kussakin on enintään 10 merkkiä.

```
40 DIM P%(10,20,5)
```

Varaa tilaa matriisille, jossa on 11 * 21 * 6 kokonaislukualkiota.

```
50 DIM P$(10,20) = 40
```

Varaa tilaa merkkijonomatriisille, jossa on 11 riviä ja 21 saraketta ja jokainen alkio on korkeintaan 40 merkkiä pitkä.

END
Lopettaa ohjelman suorituksen ja sulkee kaikki tiedostot.

EXTEND
Sallii pitkät muuttujanimet. Vrt NO EXTEND.

```
10 EXTEND
20 Bruttohinta = Nettohinta * 1.25
```

FNEND
Lopettaa monirivisen funktion määrittelyn. Katso DEF FN ... (...).

GOSUB ...

Aiheuttaa hypyn aliohjelmaan annetulla rivillä. Vrt. RETURN.

```
80 GOSUB 2000
```

NO EXTEND

Kumoaa EXTEND-käskyn vaikutuksen. Aiheuttaa sen, että vain yhdestä kirjaimesta tai yhdestä kirjaimesta ja numerosta (0:sta 9:ään) muodostuvia muuttujanimiä saa käyttää.

ON ERROR GOTO ...

Aiheuttaa hypyn virheenkäsittelyrutiiniin, mikäli ajon aikana sattuu virhe. Vertaa RESUME. ON ERROR GOTO ilman rivinumeroa kumoaa aikaisemman käskyn ON ERROR GOTO ...

```
10 ON ERROR GOTO 1000
```

ON ... GOSUB ..., ...

Aiheuttaa hypyn yhteen ilmoitetuista aliohjelmista.

```
60 ON P GOSUB 100,200,300
```

(P = 1 aiheuttaa hypyn aliohjelmaan rivillä 100, P = 2 aiheuttaa hypyn aliohjelmaan rivillä 200 jne.)

ON ... GOTO ..., ...

Aiheuttaa hypyn yhdelle ilmoitetuista ohjelmariveistä.

```
70 ON R GOTO 100,200,300
```

(R = 1 aiheuttaa hypyn riville 100, R = 2 aiheuttaa hypyn riville 200 jne.)

REM ... (vaihtoehtoisesti !...)

Käytetään ohjelman kommenteissa. Kaksoispiste voidaan jättää pois !-merkin edestä.

```
80 REM *** ALIOHJELMA ***
90 A = 5 ! alkuarvo
```

RESUME ...

Aiheuttaa paluun virherutiinista ilmoitetulle ohjelmariville. Jos rivinumero jätetään ilmoittamatta, suoritetaan paluu riville, jolla virhe tapahtui.

```
80 RESUME 10
90 RESUME
```

RETURN

Aiheuttaa paluun aliohjelmasta vastaavaa GOSUB-lausetta seuraavaan lauseeseen.

RETURN ...

Palauttaa monirivisen funktion arvon. Vrt DEF FN ... (...)

STOP

Lopettaa ohjelman ajon.

Komennot

ERASE ...
Poistaa ilmoitetut ohjelmarivit. Seuraavia muotoja voi käyttää:

ERASE 40—80	(Poistaa rivit välillä 40—80.)
ERASE —80	(Poistaa rivit riviin 80 saakka.)
ERASE 40—	(Poistaa rivit rivistä 40 alkaen.)

MERGE ... ...
Siirtää tekstimuotoisen ohjelman (talletettu komennolla LIST ...) kasetilta tai levykkeeltä ABC800:aan poistamatta muistissa jo olevaa ohjelmaa. (Jos muistissa olevassa ohjelmassa on käytetty samoja rivinumeroita, ne korvautuvat ladattavan ohjelman riveillä.)

```
MERGE OHJELMA2
MERGE DR1: OHJELMA 2
```

Harjoitustehtäviä

* 7.1 Varusta sivun 57 ohjelma aloitusosalla, josta käy ilmi mm. ohjelman nimi ja ohjelmoija.

* 7.2 Lisää sivun 57 ohjelmaan virherutiini, ettei ohjelman ajo pääty virheelliseen tietojen syöttöön.

* 7.3 Täydennä sivun 42 ohjelmaa siten, että sille voidaan syöttää 200 enintään 40-merkkistä nimeä ja vastaavat puhelinnumerot, jotka voivat olla korkeintaan 20-merkkisiä.

* 7.4 Ratkaise harjoitustehtävä 4.4 käyttämällä pitkiä muuttujanimiä.

* 7.5 Tässä ohjelmassa lasketaan liikevaihtoveron sisältävä hinta käyttämällä apuna määriteltyä funktiota. Aja ohjelma! Talleta sitten funktionmäärittely levykkeelle erillisenä ohjelmana.

```
10 REM OHJELMA                          : L7H5
20 EXTEND
30 PRINT CHR$(12)
40 PRINT "LIIKEVAIHTOVERON LASKENTA"
50 PRINT
60 INPUT "HINTA ILMAN LVV:TÄ            : "Hinta
70 PRINT "HINTA LVV:N KANSSA            : " FNLvv(Hinta)
80 PRINT
90 PRINT
100 GOTO 50
110 REM ------- LVV.DEF -------
120 DEF FNLvv(Hinta)
130  Lvv=.2151
140  Loppuhinta=Hinta+Hinta*Lvv
150  RETURN Loppuhinta
160 FNEND
```

* 7.6 Mitä tapahtuu, jos ohjelma VARASTO puuttuu levykkeeltä, kun allaoleva ohjelma ajetaan? Kuten huomaat, ei virherutiini toimi tyydyttävästi, vaan ajo täytyy keskeyttää tietokoneen tarvekeskeytyksellä. Muunna ohjelmaa siten, että virheiden käsittely sujuu moitteettomasti.

```
100 REM OHJELMA                         : L7H6
110 ON ERROR GOTO 110
120 REM . . .
130 REM . . .
140 CHAIN "VARASTO"
145 END
150 REM ------- VIRHEEN KÄSITTELIJÄ -------
160 PRINT
170 PRINT "TIEDOSTO EI OLE LEVYKKEELLÄ"
180 RESUME
```

7.7 Kansainvälisessä satelliittien laukaisukeskuksessa käytetään tätä ohjelmaa, joka sisältää muunnetun version ohjelmasta VIIVYTYS.DEF. Muuta ohjelmarivejä 160 ja 190 siten, että saadaan vaihtelevia viivytysaikoja.

```
100 REM OHJELMA                                    : L7H7
110 EXTEND
120 PRINT CHR$(12)
130 INPUT "KENELLE ILMOITETAAN ? "V$
140 FOR N=1 TO 20
150  PRINT CHR$(12)
160  Odota=FNViive(100)
170  PRINT "TIEDOTUS ODOTTAA , " RED V$ " "
180  PRINT "OTA YHTEYS NEUVONTAAN"
190  Odota=FNViive(2000)
200 NEXT N
210 GOTO 120
220 REM ------- VIIVE.DEF ----------
230 DEF FNViive(Aika)
240  FOR Kierros=1 TO Aika : Next Kierros
250  RETURN Eimitään
260 FNEND
```

7.8 Kirjoita laskentaohjelma, joka osaa yhteen-, vähennys-, kerto- ja jakolaskua. Ohjelmalle syötetään kaksi lukua ja laskentatapa pitäisi voida valita tällaisesta menusta:

```
LASKENTAOHJELMA

VAIHTOEHDOT OVAT:

1 YHTEENLASKU
2 VÄHENNYSLASKU
3 KERTOLASKU
4 JAKOLASKU
5 OHJELMAN LOPETUS

MINKÄ VAIHTOEHDON VALITSET ?
```

Käytä aliohjelmia ja virherutiinia.

8

Tiedostojen käsittely

Datatiedostot

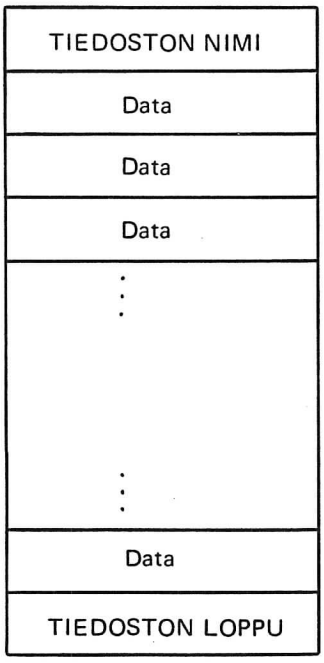
Olemme aikaisemmin näyttäneet, miten ohjelmia talletetaan levykkeelle ja kasetille. Samasta syystä kuin halutaan tallettaa ohjelmia apumuistiin, halutaan usein tallettaa myös dataa ohjelman ajosta toiseen. Tällaista levylle tai kasettinauhalle talletettua dataa sanotaan datatiedostoksi.

Lyhyt saantiaika on tärkeätä datatiedostojen kanssa työskenneltäessä. Tämän vuoksi levy-yksiköllä on huomattavia etuja kasettinauhuriin verrattuna. Kasettinauhuria käytetään datan talletukseen vain erikoistapauksissa ja siksi käsittelemme tässä luvussa vain levykkeelle talletettuja datatiedostoja.

Peräkkäistiedoston luonti

Peräkkäistiedosto voidaan määritellä sarjana tietueita, jotka on talletettu peräkkäisessä järjestyksessä mahdollisesti jonkin avaimen mukaan lajiteltuina. Ensimmäisenä esimerkkinä muodostamme peräkkäistiedoston, johon talletetaan tietoja 5 tuotteesta varastossa. Kustakin tuotteesta talletetaan 3 tietoa, nimittäin tuotenumero, tuotteen nimi ja määrä varastossa.

Apumuistiin talletettavalle datatiedostolle täytyy antaa nimi aivan kuten ohjelmiakin talletettaessa. Tiedoston loppua osoittamaan käytetään tiedoston lopetusmerkkiä. Datatiedosto on siis näin organisoitu:



Sijoita käyttöä varten alustettu levyke levyasemaan, ja talleta seuraava ohjelma levykkeelle nimellä "KIRJOITA1".

```
100 REM OHJELMA                                  : KIRJ1
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "KIRJOITETAAN 15 tietoa"
140 PRINT "PERÄKKÄISTIEDOSTOON VARASTO1.DAT"
150 PRINT
160 PREPARE "VARASTO1.DAT" AS FILE 1
170 PRINT
180 FOR Tuotenumero=1 TO 15
190  REM
200  REM ------- SYÖTTÖ ----------
210  INPUT "TUOTENUMERO                          : "Tuotenumero
220  INPUT "TUOTENIMI                            : "Tuotenimi$
230  INPUT "MÄÄRÄ VARASTOSSA                     : "Määrä
240  REM
250  REM ------- TIEDOSTOON KIRJOITUS ----
260  PRINT #1, Tuotenumero
270  PRINT #1, Tuotenimi$
280  PRINT #1, Määrä
290  REM ------------------------
300  PRINT
310 NEXT Tuotenumero
320 CLOSE 1
330 PRINT "AJO PÄÄTTYNYT"
340 END
```

Aja sitten ohjelma ja anna kaikille tuotteille tiedot, jotka talletetaan tiedostoon VARASTO1.DAT.

```
KIRJOITETAAN 15 TIETOA
PERÄKKÄISTIEDOSTOON VARASTO1.DAT

TUOTENUMERO:  1
TUOTENIMI:    VASARA
MÄÄRÄ VARASTOSSA: 12345

TUOTENUMERO:  2
TUOTENIMI:    KIRVES
MÄÄRÄ VARASTOSSA: 4567

TUOTENUMERO:  3
TUOTENIMI:    RUUVIMEISSELI
MÄÄRÄ VARASTOSSA: 789
```

Kun ohjelmarivi 160 suoritetaan, muodostetaan tiedosto "VARASTO1.DAT".

```
160 PREPARE "VARASTO1.DAT" AS FILE 1
```

Tiedoston muodostaminen tarkoittaa tässä sitä, että tietokone etsii levykkeeltä vapaata tilaa ja kirjoittaa tiedoston nimen levykkeelle. Lisäksi annetaan tiedostolle "VARASTO1.DAT" numero 1, jolla ohjelmassa myöhemmin viitataan tiedostoon.

Dataa kirjoitetaan tiedostoon, kun rivit 260—280 suoritetaan.

```
260 PRINT #1,Tuotenumero
270 PRINT #1,Tuotenimi$
280 PRINT #1,Lukumäärä
```

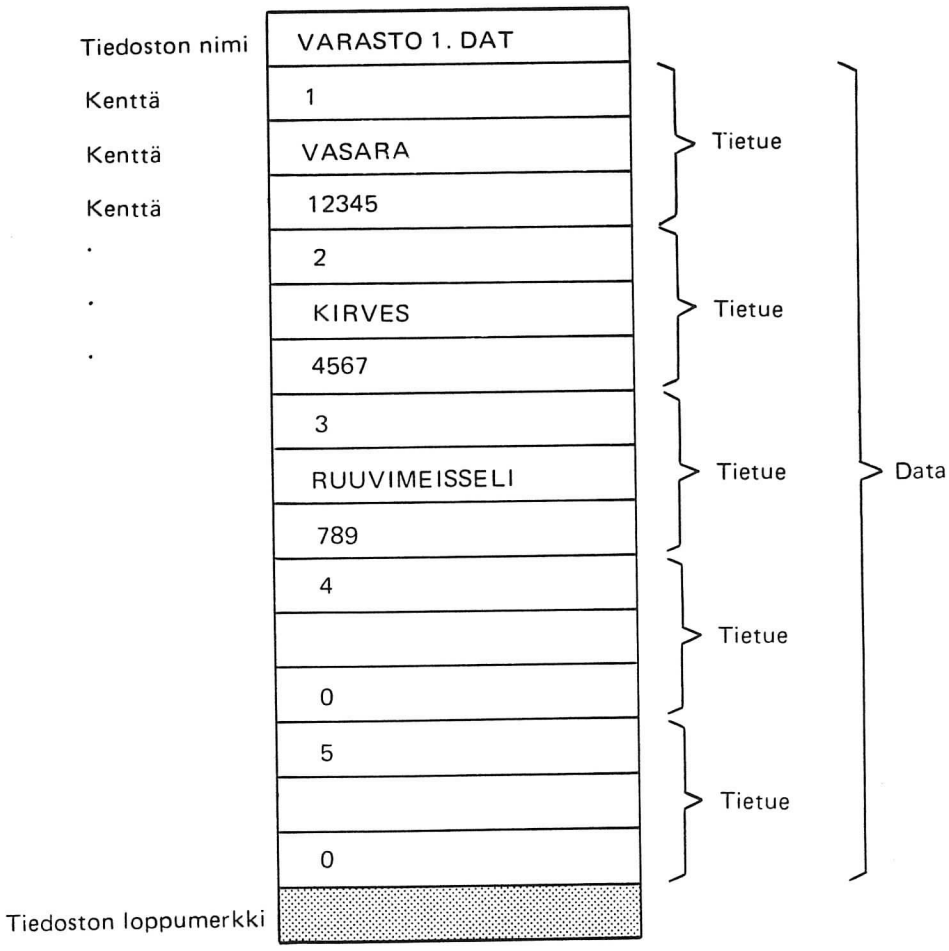
Tiedostoon kirjoitetaan PRINT-käskyllä, johon on liitetty numeromerkki # ja tiedoston numero. Näille riveille kirjoitetaan siten lukumuuttujan Tuotenumero, merkkijonomuuttujan Tuotenimi$ ja lukumuuttujan Lukumäärä arvot peräkkäisessä järjestyksessä. Koska nämä ohjelmarivit ovat silmukassa, joka suoritetaan 5 kertaa, kirjoitetaan tiedostoon siten 15 datakenttää.

Tiedostoon kirjoitus lopetetaan käskyllä CLOSE, johon on liitetty tiedoston numero.

```
320 CLOSE 1
```

Kun rivi 320 suoritetaan, kirjoittaa tietokone levykkeelle tiedoston lopetusmerkin osoittamaan tiedoston "VARASTO1.DAT" loppua.

Tiedostojen käsittelyn yhteydessä käytetään termejä tietue ja kenttä kuvaamaan tiedostoon talletettavia tietoja.



Tässä esimerkissä tietueen muodostavat yhden tuotteen tiedot. Kukin tietue puolestaan muodostuu kolmesta kentästä, nimittäin tuotteen numerosta, nimestä ja määrästä.

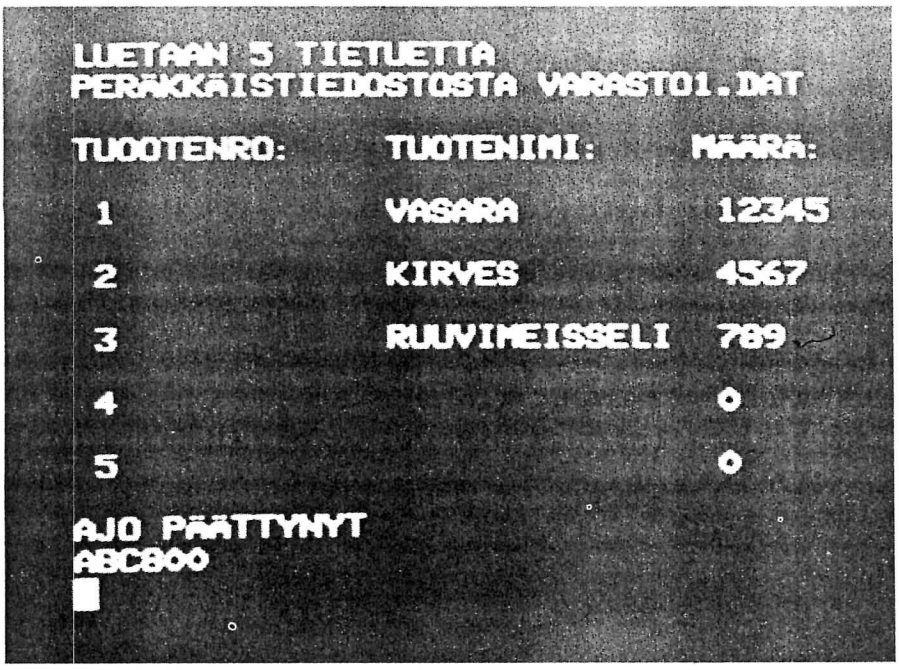
Peräkkäistiedoston lukeminen

Talleta seuraava ohjelma levykkeelle nimellä "LUE1".

```
100 REM OHJELMA                          : LUE1
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "LUETAAN 5 TIETUETTA"
140 PRINT "PERÄKKÄISTIEDOSTOSTA VARASTO1.DAT"
150 PRINT
160 OPEN "VARASTO1.DAT" AS FILE 1
170 REM
180 PRINT "TUOTENRO: ","TUOTENIMI: ","MÄÄRÄ: "
190 PRINT
```

```
200 FOR Tietuenumero = 1 TO 5
210  REM
220  REM ------- TIEDOSTON LUKU -------
230  INPUT #1, Tuotenumero
240  INPUT #1, Tuotenimi$
250  INPUT #1, Määrä
260  REM
270  REM ------- TULOSTUS ----------
280  PRINT Tuotenumero,
290  PRINT Tuotenimi$,
300  PRINT Määrä
310  REM ------------------------
320  PRINT
330 NEXT Tietuenumero
340 CLOSE 1
350 PRINT "AJO PÄÄTTYNYT"
360 END
```

Ohjelman ajon jälkeen voi kuvaruutu näyttää tältä.



Kun ohjelmarivi 160 suoritetaan, etsii tietokone tiedoston "VARASTO1.DAT" ja antaa sille numeron 1.

```
160 OPEN "VARASTO1.DAT" AS FILE 1
```

Tiedoston lukeminen tapahtuu sitten kenttä kerrallaan, kun rivit 230—250 suoritetaan.

```
230 INPUT #1,Tuotenumero
240 INPUT #1,Tuotenimi$
250 INPUT #1,Lukumäärä
```

Data siirretään tiedostosta ilmoitettuun muuttujaan käskyllä INPUT, johon on liitetty numeromerkki # ja tiedoston numero. Kun talletettu tieto on merkkijono, täytyy tietenkin INPUT-lauseen muuttujan olla merkkijonomuuttuja.

Kun 5 tietuetta on luettu tiedostosta, suljetaan tiedosto rivillä 340.

```
340 CLOSE 1
```

Kahdessa ensimmäisessä esimerkkiohjelmassa on tiedostoon kirjoitettu ja tiedostosta luettu etukäteen määrätty määrä tietueita. Seuraavissa esimerkeissä näytämme, miten käsitellään peräkkäistiedostoja, joissa on mielivaltainen määrä tietueita.

Kirjoitetaan haluttu määrä tietueita

Koska kirjoitettavien tietueiden määrä ei useimmiten ole tiedossa ohjelmaa laadittaessa, täytyy ohjelma muotoilla siten, että tietueiden lukumäärä voidaan määrätä ajon aikana. Ohjelmalla KIRJOITA2 voidaan kirjoittaa toivottu määrä tietueita tiedostoon VARASTO2.DAT.

```
100 REM OHJELMA                    : KIRJ2
110 EXTEND
120 PRINT CHR$(12)
130 PRINT 'KIRJOITETAAN HALUTTU MÄÄRÄ TIETUEITA"
140 PRINT "PERÄKKÄISTIEDOSTOSTA VARASTO2.DAT"
150 PRINT
160 PREPARE "VARASTO2.DAT" AS FILE 1
170 PRINT
180 WHILE Vastaus$<>"EI"
```

```
190  REM ------- SYÖTTÖ ----------
200  INPUT "TUOTENUMERO"              : "Tuotenumero
210  INPUT "TUOTENIMI                 : "Tuotenimi$
220  INPUT "MÄÄRÄ VARASTOSSA          : "Määrä
230  PRINT
240  INPUT "JATKETAANKO SYÖTTÖÄ ? (KYLLÄ/EI) "Vastaus$
250  PRINT
260  REM ------- TIEDOSTOON KIRJOITUS ----
270  PRINT #1, Tuotenumero
280  PRINT #1, Tuotenimi$
290  PRINT #1, Määrä
300  REM ------------------------
310 WEND
320 CLOSE 1
330 PRINT "AJO PÄÄTTYNYT"
340 END
```

Kappaleita SYÖTTÖ ja TIEDOSTON KIRJOITUS toistetaan kunnes muuttuja Vastaus$ rivillä 240 saa arvon EI. Tiedostoon VARASTO2.DAT kirjoitettavien tietueiden määrää rajoittaa siksi ainoastaan levykkeeltä löytyvän vapaan tilan määrä.

Luetaan tiedostoa, jonka pituus ei ole tunnettu

Jokainen tiedosto päättyy loppumerkkiin. Luettaessa tuntemattoman mittaista tiedostoa käytetään tätä loppumerkkiä hyväksi. Kun yritetään lukea tietuetta tämän loppumerkin jälkeen, aiheutuu "virhe, joka voidaan tavalliseen tapaan käsitellä käskyllä ON ERROR GOTO ... Seuraavalla ohjelmalla voidaan lukea tiedostoa VARASTO2.DAT riippumatta siitä, miten monta tietuetta tiedostossa on.

```
100 REM OHJELMA                       : LUE2
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "LUETAAN KAIKKI TIETUEET"
140 PRINT "PERÄKKÄISTIEDOSTOSTA VARASTO2.DAT"
150 PRINT CUR(23,0); "(Paina RETURN, kun haluat uuden tietueen)";
160 PRINT CUR(6,0);
170 PRINT
```


Kotkan
Kaupunginkirjasto

```
180 OPEN "VARASTO2.DAT" AS FILE 1
190 REM
200 PRINT "TUOTENRO: ","TUOTENIMI: ","MÄÄRÄ: "
210 PRINT
220 REM
230 REM ------- TIEDOSTON LUKU -------
240 ON ERROR GOTO 400
250 INPUT #1, Tuotenumero
260 INPUT #1, Tuotenimi$
270 INPUT #1, Määrä
280 REM
290 REM ------- TULOSTUS ----------
300 PRINT Tuotenumero,
310 PRINT Tuotenimi$,
320 PRINT Määrä
330 GET Vastaus$
340 REM ------------------------
350 GOTO 220
360 CLOSE 1
370 PRINT
380 PRINT "AJO PÄÄTTYNYT"
390 END
400 REM ------- VIRHEEN KÄSITTELIJÄ -------
410 RESUME 360
```

Luettaessa dataa tiedostosta täytyy tietää, miten tieto on talletettu tiedostoon. Tietueiden lukumäärää ei tarvitse tietää, sen sijaan kunkin tietueen rakenne on tiedettävä. Toisin sanoen kunkin tietueen kenttien lukumäärä ja kunkin datakentän tyyppi on oltava selvillä. Kappale TIEDOSTON LUKU on siksi saman näköinen kuin ohjelmassa "LUE1".

Kappaleita TIEDOSTON LUKU ja TULOSTUS toistetaan, kunnes tiedoston loppumerkki aiheuttaa virheen luettaessa. Kun virhe sattuu, siirrytään rutiiniin VIRHEEN KÄSITTELIJÄ, jossa tiedosto suljetaan ja ajo lopetetaan.

Kun rivi 330 suoritetaan, odottaa tietokone syöttöä näppäimistöstä vastaavalla tavalla kuin INPUT-lauseessa.

```
330 GET Vastaus$
```

Kun käytetään käskyä GET ..., jatketaan ohjelman suoritusta heti kun jotain näppäintä on painettu. Painettua näppäintä vastaava merkki asetetaan GET-lauseen merkkijonomuuttujan arvoksi, mutta sitä ei tulosteta kuvaruudulle.

Ohjelmarivin 330 tarkoitus on tässä vain tulostuksen keskeyttäminen kunkin tietueen jälkeen. Muuttujan Vastaus$ arvoa ei siis käytetä tässä mihinkään.

Funktionäppäinten käyttäminen

Tietokoneen näppäinpöydässä on kahdeksan funktionäppäintä, jotka on merkitty nimillä PF1, ..., PF8. Kun funktionäppäimiä käytetään tietojen syötössä, generoituu ''virhe'', joka voidaan käsitellä käskyllä ON ERROR GOTO ... tavalliseen tapaan. Näille näppäimille voi näin antaa ohjelmassa erityisominaisuuksia ajon ajaksi.

Seuraava ohjelma näyttää, miten funktionäppäimiä voi käyttää syötön päättämiseen ja siten lopettamaan tiedoston ''VARASTO2.DAT'' kirjoittamisen.

Koska tämä ohjelma muodostaa samannimisen tiedoston kuin muodostettiin ohjelmalla ''KIRJOITA2'', tullaan aikaisempi tiedosto ''VARASTO2.DAT'' tuhoamaan.

```
100 REM OHJELMA                    : KIRJ3
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "KIRJOITETAAN HALUTTU MÄÄRÄ TIETUEITA"
140 PRINT "PERÄKKÄISTIEDOSTOON VARASTO2.DAT"
150 PRINT
160 PRINT "Syöttö keskeytetään"
170 PRINT "funktionäppäimellä PF1"
180 REM
190 PREPARE "VARASTO2.DAT" AS FILE 1
200 PRINT
```

```
210 REM ------- SYÖTTÖ ----------
220 ON ERROR GOTO 380
230 INPUT "TUOTENUMERO                    : "Tuotenumero
240 INPUT "TUOTENIMI                      : "Tuotenimi$
250 INPUT "MÄÄRÄ VARASTOSSA               : "Määrä
260 PRINT
270 PRINT
280 REM ------- TIEDOSTOON KIRJOITUS ----
290 PRINT #1, Tuotenumero
300 PRINT #1, Tuotenimi$
310 PRINT #1, Määrä
320 REM ----------------------
330 GOTO 210
340 CLOSE 1
350 PRINT : PRINT
360 PRINT "AJO PÄÄTTYNYT"
370 END
380 REM ------- VIRHEEN KÄSITTELIJÄ ————————
390 IF ERRCODE=53 THEN RESUME 340
400 PRINT " VIRHE !"
410 RESUME
```

Ohjelmakappaleista SYÖTTÖ ja TIEDOSTON KIRJOITUS muodostuvan silmukan suoritus voidaan lopettaa ainoastaan generoimalla "virhe". Virhe voi aiheutua joko funktionäppäimen painamisesta tai virheellisestä syötöstä.

Kuten sivun 10 ohjelmointikortista käy ilmi, vastaa jokaista "virhettä" virhekoodi, joka on luku. Muuttujan ERRCODE arvoksi asetetaan automaattisesti viimeisen ajon aikaisen virheen virhekoodi. Muuttujan ERRCODE avulla voidaan siten määritellä, mikä "virhe" on sattunut.

Virhekoodi 53 tarkoittaa, että jotain funktionäppäintä on painettu. Ohjelmarivi 390 määrää, että funktionäppäimen painamisen jälkeen ohjelman suoritusta jatketaan riviltä 340. Jos funktionäppäimille halutaan antaa eri merkityksiä, täytyy laatia rutiini selvittämään, mitä näppäintä painettiin. Tällainen rutiini on kuvattu osassa HARJOITUSTEHTÄVIÄ.

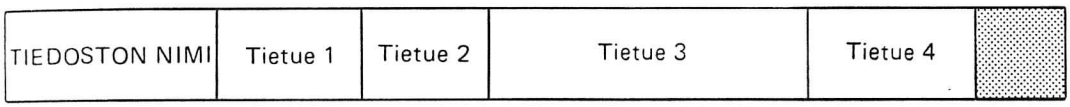
Muuntyyppisten virheiden, esim. virheellisen syötön, tapahtuessa jatketaan ohjelman suoritusta tavalliseen tapaan siitä, missä virhe sattui.

Ohjelma ''KIRJOITA3'' luo samannimisen ja samanlaisen tiedoston kuin ohjelma ''KIRJOITA2''. Sen vuoksi kummassakin tapauksessa tiedostoa ''VARASTO2.DAT'' voidaan lukea ohjelmalla ''LUE2''.

Peräkkäistiedostot — suorasaantitiedostot

Kuten peräkkäistiedostoja käsitelleistä esimerkeistä on käynyt ilmi, voidaan peräkkäistiedostoa kuvata tällä tavalla:

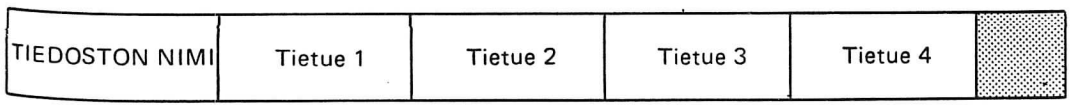
- Data kirjoitetaan aina peräkkäisessä järjestyksessä tiedoston alusta alkaen.
- Dataa luetaan aina peräkkäisessä järjestyksessä tiedoston alusta alkaen.
- Tietueen pituus määräytyy tiedostoon talletettavan datan perusteella.
- Tiedoston pituus määräytyy kirjoituksen yhteydessä ja riippuu tietueiden pituuksista ja lukumäärästä.



Peräkkäistiedosto

Datatiedoston voi vaihtoehtoisesti organisoida suorasaantitiedostoksi. Suorasaantisen tiedoston etu on esimerkiksi se, että mielivaltainen tietue voidaan lukea ilman, että kaikki sitä edeltävät tietueet luetaan. Suorasaantisuus kuitenkin edellyttää, että tietueiden lukumäärä ja pituus määrätään tiedostoa luotaessa. Suorasaantitiedoston ominaisuudet ovat seuraavat:

- Kaikki tietueet ovat yhtä pitkiä.
- Tiedosto täytyy alustaa, eli kaikille tietueille on varattava tila ennen kuin tiedostoa voi käyttää.
- Tietueita voidaan kirjoittaa ja vastaavasti lukea mielivaltaisessa järjestyksessä.



Suorasaantitiedosto

Toisin sanoen suorasaantitiedosto voidaan määritellä jollekin ulkoiselle muistilaitteelle talletettujen tietueiden joukkona, josta tietueita voidaan lukea ja johon niitä voi kirjoittaa mielivaltaisessa järjestyksessä.

Suorasaantitiedoston alustus

Ohjelma ALUSTA1 varaa levykkeeltä tilaa tiedostolle VARASTO3.DAT. Tiedosto koostuu 21 tietueesta, joissa kussakin on 40 merkkiä.

```
100 REM OHJELMA                          : ALUSTA1
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "ALUSTETAAN SUORASAANTITIEDOSTO, JOSSA ON"
140 PRINT "21 TIETUETTA JA KUNKIN TIETUEEN PITUUS 40 MERKKIÄ"
150 PREPARE "VARASTO3.DAT" AS FILE 1
160 FOR Tietuenumero = 0 TO 20
170  PUT #1, SPACE$(40)
180 NEXT Tietuenumero
190 CLOSE 1
200 PRINT
210 PRINT "ALUSTUS PÄÄTTYNYT"
220 END
```

Tavalliseen tapaan annetaan tiedoston nimi ja numero ohjelmarivillä 150.

Suorasaantitiedostoon kirjoitetaan käskyllä PUT, johon liitetään numeromerkki #, tiedoston numero ja tiedostoon kirjoitettava merkkijono. Alustuksessa tiedostosta varataan vain tilaa tulevia tietueita varten ja siksi tietueiden paikalle kirjoitetaankin vain 40 välilyöntiä.

```
170 PUT #1,SPACE$(40)
```

Funktio SPACE(40) antaa 40 välilyönnistä muodostuvan merkkijonon. Joka kerta kun rivi 170 suoritetaan, kirjoitetaan siis 40 välilyönnin muodostama tyhjä tietue tiedostoon.

Alustus päättyy, kun rivillä 190 tiedosto suljetaan. Tiedosto on tällöin alustettu ja valmis käyttöä varten.

Tietueiden kirjoitus suorasaantitiedostoon

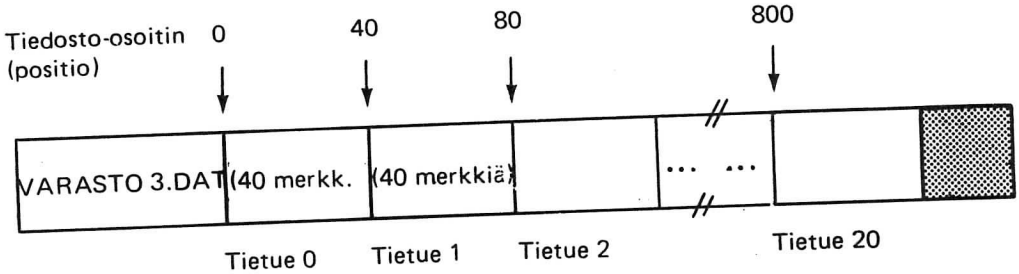
Seuraava ohjelma näyttää, miten tietueita kirjoitetaan suorasaantitiedostoon mielivaltaisessa järjestyksessä. Kuten ohjelmassa KIRJOITA3 tässäkin syöttö lopetetaan funktionäppäimillä.

```
100 REM OHJELMA                          : KIRJ4
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "KIRJOITETAAN TIETUEITA SUORASAANTITIEDOSTOON"
140 PRINT "VARASTO3.DAT (Lopetus näppäimellä PF1)"
150 PRINT
160 OPEN "VARASTO3.DAT" AS FILE 1
170 ON ERROR GOTO 340
180 REM
190 REM ------- SYÖTTÖ ----------
200 PRINT
210 INPUT "TIETUENUMERO                  : "Tietuenumero
220 IF Tietuenumero>20 THEN GOTO 210
230 PRINT
240 PRINT "TUOTENUMERO                   : "Tietuenumero
250 INPUT "TUOTENIMI                     : "Tuotenimi$
260 REM
270 REM ------- TIEDOSTOON KIRJOITUS ----
280 PRINT #1, Tuotenumero
290 PRINT #1, Tuotenimi$
300 REM ------------------------
310 GOTO 190
320 CLOSE 1
330 PRINT ' '
340 PRINT "AJO PÄÄTTYNYT"
350 END
360 REM ------- VIRHEEN KÄSITTELIJÄ -------
370 RESUME 320
```

Kun ohjelmarivi 160 suoritetaan, avataan aikaisemmin alustettu tiedosto VARASTO3.DAT ja annetaan sille tiedostonumero 1. Rivillä 220 tarkistetaan, että syötetty tietuenumero on sallittu. Tälle tiedostolle tietuenumerot 0—20 ovat sallittuja.

Kuten riviltä 240 käy ilmi, annamme tuotenumeron olla sama kuin tietuenumero yksinkertaisuuden vuoksi.

Koska suorasaantitiedoston kirjoitus voi tapahtua valinnaiseen paikkaan, täytyy toivottu positio ilmoittaa ennen kirjoittamista. Positiossa 0 on tiedoston ensimmäinen merkki, toinen merkki on positiossa 1 jne. Erään tietueen ensimmäisen merkin positio selviää seuraavasta kuviosta.



Positio, josta kirjoittaminen aloitetaan, ilmoitetaan käskyllä POSIT # ..., ... Rivillä 280 annetaan kyseessä olevan tietueen ensimmäisen merkin positio.

```
280 POSIT #1,Tietuenumero*40
```

Kun tämä rivi on suoritettu, muodostuu kyseinen tietue osittain tuotenimen merkeistä ja osittain joukosta välilyöntejä. Tuotenimen kirjainten lukumäärän ja välilyöntien lukumäärän summa on kussakin tietueessa 40.

Kun kirjoittaminen lopetetaan, suljetaan tiedosto tavalliseen tapaan käskyllä CLOSE ...

```
290 PUT #1,Tuotenimi$
```

Kun positio on määrätty, voidaan tiedostoon kirjoittaa merkkijono käskyllä PUT # ..., ...

Suorasaantitiedoston tietueiden lukeminen

Luettaessa suorasaantitiedostosta tiedoston avaaminen ja position määrääminen tapahtuu kuten kirjoitettaessakin.

Käskyllä GET # ..., ... COUNT ... luetaan joukko merkkejä tiedostosta.

```
260 GET #1,Tuotenimi$ COUNT 40
```

Tätä käskyä suoritettaessa siirretään 40 merkkiä tiedostosta numero 1 merkkijonomuuttujaan Tuotenimi$. Koska tiedosto-osoitin on asetettu osoittamaan tietueen ensimmäiseen merkkiin, siirretään näin siis kokonainen tietue tiedostosta.

Seuraava ohjelma lukee annetun tietueen tiedostosta VARASTO3.DAT.

```
100 REM OHJELMA                          : LUE3
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "LUETAAN HALUTTU TIETUE SUORASAANTITIEDOSTOSTA"
140 PRINT "VARASTO3.DAT (Lopetus näppäimellä PF1)"
150 PRINT
160 OPEN "VARASTO3.DAT" AS FILE 1
170 ON ERROR GOTO 360
180 REM
190 REM ------- SYÖTTÖ ----------
200 PRINT
210 INPUT "TUOTENUMERO                   : "Tietuenumero
220 IF Tietuenumero>20 THEN 210
230 REM
240 REM ------ TIEDOSTON LUKU -------
250 POSIT #1, Tietuenumero*40
260 GET #1, Tuotenimi$ COUNT 40
270 REM
280 REM ------- TULOSTUS ----------
290 PRINT Tuotenimi$
300 PRINT "------------------------"
310 GOTO 190
320 CLOSE 1
330 PRINT
340 PRINT "AJO PÄÄTTYNYT"
350 END
360 REM ------- VIRHEEN KÄSITTELIJÄ ----
370 RESUME 320
```

Luettaessa ja kirjoitettaessa suorasaantitiedostoon on data aina merkkijonomuodossa. Tietysti myös lukuja voi tallettaa suorasaantitiedostoon, mutta tällöin tarvitaan joskus konvertointia lukujen ja merkkijonojen välillä. Esimerkkejä tällaisista muunnoksista on seuraavissa ohjelmissa.

Useita kenttiä tietueessa

Ensimmäinen suorasaantitiedoston käsittelyohjelma oli laadittu siten, että kukin tietue muodostui ainoastaan yhdestä kentästä, nimittäin tuotteen nimestä. Tavallisesti tietueessa on useita kenttiä. Jäljellä olevissa kolmessa esimerkissä annamme siksi kunkin tietueen muodostua kahdesta kentästä: tuotteen nimestä ja määrästä varastossa. Näytämme myös millaisia muunnoksia vaaditaan tallettaessa lukuja suorasaantitiedostoon. Viimeisessä esimerkissä näytämme, miten tietueita päivitetään.

Kun ajat tämän ohjelman, tallentuu 21 tuotteen nimeä ja niiden määrät varastossa tiedostoon VARASTO3.DAT. Ennen ohjelman ajoa täytyy tiedosto alustaa ohjelmalla ALUSTUS1.

```
100 REM OHJELMA                                    : KIRJ5
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "KIRJOITETAAN 21 TIETUETTA"
140 PRINT "SUORASAANTITIEDOSTOON VARASTO3.DAT"
150 PRINT "(Lopetus näppäimellä PF1)"
160 PRINT
170 Kentänpituus1 = 36 : REM TUOTENIMIKENTTÄ
180 Kentänpituus2 = 4 : REM MÄÄRÄKENTTÄ
190 Tietueenpituus = Kentänpituus1 + Kentänpituus2
200 OPEN "VARASTO3.DAT" AS FILE 1
210 ON ERROR GOTO 430
220 FOR Tietuenumero = 0 TO 20
230  REM
240  REM ------- SYÖTTÖ ----------
250  PRINT
260  PRINT "TUOTENUMERO                    : "Tietuenumero
270  INPUT "TUOTENIMI                      : "Tuotenimi$
280  INPUT "MÄÄRÄ VARASTOSSA               : "Määrä
290  REM
300  REM ------- TIEDOSTOON KIRJOITUS ----
310  Tietueenpaikka = Tietuenumero * Tietueenpituus
320  POSIT #1, Tietueenpaikka
330  PUT #1, Tuotenimi$
340  POSIT #1, Tietueenpaikka + Kentänpituus1
350  Määrä$ = CVTF$(Määrä)
360  PUT #1, Määrä$
370  REM ------------------------
380 NEXT Tietuenumero
390 CLOSE 1
400 PRINT
```

```
410 PRINT "AJO PÄÄTTYNYT"
420 END
430 REM ------- VIRHEEN KÄSITTELIJÄ -------
440 IF ERRCODE=53 THEN RESUME 390
450 PRINT " VIRHE !"
460 RESUME
```

Selvittääksemme tämän ohjelman toimintaa esitämme nyt muutamien ohjelmarivien merkityksen.

Rivinumero	**Rivin merkitys**
170	Tuotenimen kenttä muodostuu 36 merkistä.
180	Määräkenttä muodostuu 4 merkistä.
190	Tietueen pituus on 40 merkkiä kuten ennenkin.
200	Suorasaantitiedosto VARASTO3.DAT avataan kirjoitusta varten.
210	Virheellinen syöttö ja funktionäppäinten käyttö aiheuttavat hypyn rutiiniin VIRHEIDEN KÄSITTELY.
220, 380	Tietueiden lukumäärä tiedostossa on 21.
270, 280	Nimi ja määrä syötetään kullekin tuotteelle.
310	Kyseessä olevan tietueen ensimmäisen merkin positio lasketaan.
320	Kyseessä olevan tietueen ensimmäisen kentän positio asetetaan.
330	Tuotenimi kirjoitetaan tiedostoon.
340	Kyseisen tietueen toisen kentän positio asetetaan.

350	Lukumuuttujan Määrä arvo konvertoidaan (engl. ConVerT) 4 merkin merkkijonoksi erikoisfunktiolla CVTF$(...).
360	Määrää vastaava merkkijono kirjoitetaan tiedostoon.
390	Tiedosto suljetaan.
440	Virhekoodi 53 saadaan funktionäppäimen painamisesta ja se aiheuttaa tiedoston sulkemisen ja ajon lopettamisen.
450, 460	Virheellinen syöttö käsitellään tavalliseen tapaan.

Suorasaantitiedoston lukemiseen tarkoitettu ohjelma muistuttaa hyvin paljon vastaavaa kirjoitusohjelmaa. Tämä johtuu siitä, että käskyllä OPEN "..." AS FILE ... suorasaantitiedosto avataan sekä kirjoittamista että lukemista varten. Lisäksi käytetään käskyä POSIT # ..., ... tiedosto-osoittimen asettamiseen sekä kirjoitettaessa käskyllä PUT # ..., ... että luettaessa käskyllä GET # ..., ... COUNT ...

Seuraavassa ohjelmassa, jolla luetaan suorasaantitiedostoa VARASTO3.DAT, ainoa uusi asia on määräkentän neljän merkin konvertoiminen.

```
300 Määrä = CVT$F(Määrä$)
```

Merkkijonomuuttujan Määrä$ neljä merkkiä muutetaan luvuksi erikoisfunktiolla CVT$F(...) ja tämä luku asetetaan lukumuuttujan Määrä arvoksi.

Tiedostosta VARASTO3.DAT tietueita lukeva ohjelma voi näyttää tältä.

```
100 REM OHJELMA                          : LUE4
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "LUETAAN 21 TIETUETTA"
140 PRINT "SUORASAANTITIEDOSTOSTA VARASTO3.DAT"
150 PRINT "(Lopetus näppäimellä PF1)"
160 PRINT
170 Kentänpituus1 = 36 : REM TUOTENIMIKENTTÄ
180 Kentänpituus2 = 4 : REM MÄÄRÄKENTTÄ
190 Tietueenpituus = Kentänpituus1 + Kentänpituus2
200 OPEN "VARASTO3.DAT" AS FILE 1
210 ON ERROR GOTO 470
220 FOR Tietuenumero = 0 TO 20
230  REM
240  REM ------- TIEDOSTON LUKU ----------
250  Tietueenpaikka = Tietuenumero * Tietueenpituus
260  POSIT #1, Tietueenpaikka
270  GET #1,Tuotenimi$ COUNT Kentänpituus1
280  POSIT #1,Tietueenpaikka + Kentänpituus1
290  GET #1,Määrä$ COUNT Kentänpituus2
300  Määrä = CVT$F(Määrä$)
310  REM
320  REM ------- TULOSTUS -------
330  PRINT
340  PRINT "TUOTENUMERO : " Tietuenumero
350  PRINT
360  PRINT "TUOTENIMI : " Tuotenimi$
370  PRINT "MÄÄRÄ VARASTOSSA : " Määrä
380  PRINT CUR(20,0);
390  INPUT "PAINA RETURN , KUN HALUAT UUDEN TIETUEEN"Vastaus$
400  PRINT CHR$(12)
410  REM ------------------------
420 NEXT Tietuenumero
430 CLOSE 1
440 PRINT
450 PRINT "AJO PÄÄTTYNYT"
460 END
470 REM ------- VIRHEEN KÄSITTELIJÄ -------
480 RESUME 420
```

Kappaletta TIEDOSTOSTA LUKEMINEN voidaan yksinkertaistaa jättämällä rivi 280 pois. Luettaessa käskyllä GET # ..., ... COUNT ... kasvatetaan nimittäin tiedosto-osoittimen arvoa automaattisesti COUNT-sanaa seuraavalla luvulla. Tämän vuoksi tiedosto-osoitin osoittaa valmiiksi oikeaan kohtaan toista kenttää luettaessa.

Suorasaantitiedoston päivitys

Rekisteri, jossa säilytetään tietoja varastossa olevista tuotteista, on hyödyllinen vain kun tiedot on helppo pitää ajan tasalla. Varaston päivitys voi olla esimerkiksi jonkin tuotteen määrän muuttaminen, tuotteen poistaminen valikoimasta tai uuden tuotenimen lisääminen varastoon.

Seuraavassa ohjelmassa on rajoituttu päivittämään vain tuotteen määrää varastossa.

```
100 REM OHJELMA                                    : PÄIVIT1
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "SUORASAANTITIEDOSTON VARASTO3.DAT"
140 PRINT "TIETUEIDEN KENTTIEN PÄIVITYS"
150 PRINT "(Lopetus näppäimellä PF1)"
160 PRINT
170 Tietueenpituus = Kentänpituus1 + Kentänpituus2
180 Kentänpituus2 = 4
190 Postlängd = Termlängd1 + Termmlängd2
200 OPEN "VARASTO3.DAT" AS FILE 1
210 ON ERROR GOTO 610
220 PRINT
230 REM ------- SYÖTTÖ ----------
240 INPUT "MIKÄ TUOTENUMERO? "Tuotenumero
250 IF Tuotenumero>20 THEN GOTO 240
260 PRINT : PRINT
270 Tietuenumero = Tuotenumero
280 REM
290 REM ------- TIEDOSTON LUKU -------
300 Tietueenpaikka = Tietuenumero*Tietueenpituus
310 POSIT #1, Tietueenpaikka
320 GET #1, Tuotenimi$ COUNT Kentänpituus1
330 GET #1, Määrä$ COUNT Kentänpituus 2
340 Määrä = CVT$F(Määrä$)
350 REM
360 REM ------- TULOSTUS ----------
370 PRINT "TUOTENUMERO :              " Tietuenumero
380 PRINT
390 PRINT "TUOTENIMI :              " Tuotenimi
400 PRINT "MÄÄRÄ VARASTOSSA :              " Määrä
410 REM
```

```
420 REM ------- PÄIVITYS ----------
430 PRINT : PRINT
440 INPUT "VARASTOMUUTOS:              "Muutos
450 Määrä = Määrä + Muutos
460 POSIT #1, Tietueenpaikka + Kentänpituus1
470 Määrä$ = CVTF$(Määrä)
480 PUT #1, Määrä$
490 PRINT
500 PRINT "UUSI MÄÄRÄ VARASTOSSA : " Määrä
510 PRINT
520 REM ------- LOPETUS ----------
530 PRINT
540 INPUT "PAINA RETURN UUTTA PÄIVITYSTÄ VARTEN"Vastaus$
550 PRINT CHR$(12)
560 GOTO 230
570 CLOSE 1
580 PRINT
590 PRINT "AJO PÄÄTTYNYT"
600 END
610 REM ------- VIRHEEN KÄSITTELIJÄ -------
620 IF ERRCODE = 53 THEN RESUME 570
630 PRINT " VIRHE !"
640 RESUME
```

Kappaleessa SYÖTTÖ annetaan haluttu tuotenumero. Haluttu tietue noudetaan tiedostosta kappaleessa TIEDOSTOSTA LUKEMINEN ja tulostetaan kuvaruudulle kappaleessa TULOSTUS.

Haluttu varastomuutos annetaan kappaleessa PÄIVITYS rivillä 440. Uusi määrä lasketaan rivillä 450. Toisen kentän positioksi asetetaan tietueen sijainnin ja ensimmäisen kentän pituuden summa. Konvertoinnin jälkeen kirjoitetaan uutta määrää vastaava merkkijono tiedostoon. Uusi määrä kirjoitetaan myös kuvaruudulle rivillä 500. Ajo päättyy funktionäppäimen painamiseen. Virheiden käsittely on samanlainen kuin aiemmissakin esimerkeissä.

Arvokkaat datatiedostot

Datatiedostoon talletettu tieto on usein korvaamatonta. Kuten tiedät, ovat levykkeet kovin herkkiä ja niille talletettu tieto voi huonoissa olosuhteissa tuhoutua. Siksi onkin tärkeätä säännöllisesti kopioida arvokkaita levykkeitä. Datatiedostoja kopioidaan samalla tavalla kuin ohjelmia, joiden kopioimista käsiteltiin luvussa 6.

Tähän saakka olemme tallettaneet ohjelmat ja datatiedostot samalle levykkeelle. Monissa sovellutuksissa on kuitenkin käytännöllisempää tallettaa datatiedostot erilliselle levykkeelle. Tämä voidaan tehdä siten, että datalevy sijoitetaan yksikköön 1 (DR1:) ja ilmoitetaan yksikön nimi ohjelmassa yhdessä tiedoston nimen kanssa, esim. "DR1: VARASTO3.DAT". Laajemmin datatiedostojen käsittelyä kuvataan lähdeteoksessa 1.

Yhteenveto

Käskyt

CLOSE ...
Sulkee ilmoitetun tiedoston (tiedostot) eli lopettaa tiedoston lukemisen tai kirjoittamisen. Jos tiedosto on avattu kirjoittamista varten, kirjoitetaan tiedostoon loppumerkki. Jos ilmoitettu tiedosto on talletettu kasettinauhalle, nauhurin moottori pysähtyy. Mikäli tiedostonumeroa ei ilmoiteta, suljetaan kaikki avatut tiedostot.

```
50 CLOSE 4
60 CLOSE 1, 4
70 CLOSE
```

GET ...
Odottaa näppäimen painamista. Tallettaa vastaanotetun merkin ilmoitettuun merkkijonomuuttujaan kirjoittamatta sitä kuvaruudulle. Kaikki merkit hyväksytään.

```
20 GET S7$
```

GET # ..., ... COUNT ...
Siirtää dataa ilmoitetusta suorasaantitiedostosta haluttuun merkkijonomuuttujaan. Siirrettävien merkkien määrä ilmoitetaan COUNT-sanan jälkeen. Jos COUNT ... jätetään pois, siirretään vain yksi merkki. Tiedostoosoitin päivittyy automaattisesti. Vrt. POSIT (...).

```
90 GET #1, S$ COUNT 6
```

INPUT # ..., ...
Siirtää dataa ilmoitetusta tiedostosta ilmoitettuun muuttujaan.

```
60 INPUT #2, R
70 INPUT #F, R$
```

KILL "... ... ..."
Poistaa ilmoitetun tiedoston levykkeeltä. Tiedostotyyppi täytyy antaa.

```
30 KILL "DR1: TESTI.BAC"
40 KILL A$+"TESTI.TXT"
```

NAME "... ... ..." AS "... ..."
Vaihtaa tiedoston nimen. Tiedostotyyppi täytyy antaa.

```
50 NAME "DR1: VANHA.BAS" AS
   "UUSI.BAS"
```

OPEN "... ... ..." AS FILE ...
Avaa ilmoitetun tiedoston lukemista/kirjoittamista varten ja antaa sille tiedostonumeron (0—255).

```
40 OPEN "DR1: OSOITE.TXT" AS FILE 3
50 OPEN "OSOITE.TXT" AS FILE 4
60 OPEN A$+".TXT" AS FILE F
```

POSIT # ..., ...
Asettaa suorasaantitiedoston tiedosto-osoittimen osoittamaan kohtaan, josta luetaan tai johon kirjoitetaan. Vertaa funktio POSIT (...) sekä käskyt GET # ..., ... ja PUT # ..., ...

PREPARE "... ... ..." AS FILE ...
Luo ja avaa tiedoston kirjoittamista varten sekä antaa sille tiedostonumeron (0—255).

```
30 PREPARE "DR1: OSOITE.TXT" AS FILE 3
40 PREPARE A$+".TXT" AS FILE F
```

PRINT # ..., ... (vaihtoehtoisesti ; # ..., ...)
Kirjoittaa ilmoitettuun tiedostoon.

```
30 PRINT #2, "SATUNNAISLUKU"
```

PUT # ..., ...
Ilmoitettuun suorasaantitiedostoon siirretään haluttu merkkijonomuuttuja. Tiedosto-osoitinta päivitetään automaattisesti. Vrt. POSIT (...).

```
50 PUT #1, S$
```

Merkkijonofunktioita

SPACE$(S)
Antaa merkkijonon, jossa on S kpl välilyöntejä.

```
20 B$ = SPACE$(253)
```

Erikoisfunktioita

CVT... ... (...)
Mahdollistaa merkkimuunnokset (konvertoinnit) luettaessa ja/tai kirjoitettaessa tiedostoon. Seuraavat muunnokset ovat mahdollisia (R% on kokonaisluku ja T on liukuluku).

CVT%$(R%)	R%	konvertoidaan	merkkijonoksi (2 tavua)
CVT$%(S$)	S$	—"—	kokonaisluvuksi
CVTF$(T)	T	—"—	merkkijonoksi (4/8 tavua)
CVT$F(S$)	S$	—"—	liukuluvuksi

```
20 PUT #1, CVT F$(T)
```

ERRCODE
Antaa viimeisimmän virheen koodin.

```
30 E9 = ERRCODE
```

Komennot

CLEAR
Nollaa kaikki muuttujat ja sulkee kaikki tiedostot.

Harjoitustehtäviä

* 8.1 Äskettäin toimintansa aloittanut lentoyhtiö ABC on suorittanut kalliin tutkimuksen 25 tavallisimman liikennelentokoneen polttoaineen kulutuksesta. Myöhempää käsittelyä varten 25 mittaustulosta talletetaan peräkkäistiedostoksi. Laadi ohjelma, joka luo tämän tiedoston.

* 8.2 Kirjoita ohjelma, joka lukee tehtävän 8.1 mukaista tiedostoa ja laskee 25 mittaustuloksen keskiarvon.

* 8.3 Muuta tehtävän 8.2 ohjelmaa siten, että sillä voidaan lukea erinimisiä tiedostoja edellyttäen, että tiedostoon on talletettu 25 lukua.

* 8.4 Kirjoita ohjelma, joka tulostaa kuvaruudulle halutun tietueen peräkkäistiedostosta VARASTO2.DAT. Halutun tuotteen numero syötetään ohjelman alussa.

* 8.5 Laadi ohjelma, joka laatii tilauslistan tiedoston VARASTO3.DAT tuotteista. Tuote otetaan tilauslistalle, jos sen määrä varastossa on pienempi tai yhtä suuri kuin 10.

* 8.6 Kirjoita yleinen ohjelma suorasaantitiedoston alustamista varten. Tiedoston nimi, tietueen pituus ja tietueiden lukumäärä annetaan ohjelman alussa.

8.7 Täydennä tehtävän 8.6 ohjelmaa siten, että sopivalla virherutiinilla huolehditaan virheestä, joka syntyy, jos tiedosto ei mahdu levykkeelle.

8.8 Alla olevaa kappaletta VIRHEEN KÄSITTELIJÄ voidaan käyttää antamaan kahdeksalle funktionäppäimelle (PF1—PF8) eri merkityksiä. Ohjelmarivin 950 mukaan tapahtuu hyppy kahdeksaan erilaiseen rutiiniin riippuen siitä, mitä funktionäppäintä painettiin.

Käytä tätä virheen käsittelijää lähtökohtana täydentäessäsi ohjelmaa LUE4 siten, että PF1:tä käytetään lopettamaan ohjelman ajo ja PF2:ta käytetään tulostamaan edellisen tietueen sisältö.

```
10 REM OHJELMA : PFNÄPPÄIN
20 EXTEND
30 ON ERROR GOTO 900
40 REM
50 REM . . .
60 REM . . .
70 END
900 REM ------- VIRHEEN KÄSITTELIJÄ -------
910 IF ERRCODE<>53 THEN PRINT : RESUME
920 Buffer=SYS(6)
930 GET Näppäin$
```

```
940 Pfnäppäin = ASCII(Näppäin$)—191
950 ON Pfnäppäin GOTO 1000,2000,3000,4000,5000,6000,7000,8000
960 REM
1000 REM -------- PF1 --------
1010 RESUME
2000 REM -------- PF2 --------
2020 RESUME
3000 REM -------- PF3 --------
3030 RESUME
4000 REM -------- PF4 --------
4040 RESUME
5000 REM -------- PF5 --------
5050 RESUME
6000 REM -------- PF6 --------
6060 RESUME
7000 REM -------- PF7 --------
7070 RESUME
8000 REM -------- PF8 --------
8080 RESUME
```

9

San[illegible]
käsi[illegible]
peru[illegible]

Tulostus kirjoittimen avulla

Kuvaruutu on epäilemättä keskeisin väline, jolla tietokone esittää käyttäjälle 'aikaansaannoksiaan'. Merkittävänä täydennyksenä on tietojen esittäminen paperilla kirjoittimen avulla. Useimpiin tietokonejärjestelmiin on mahdollista liittää jonkintyyppinen kirjoitin. Kirjoitinta voidaan käyttää sekä ohjelmatekstien dokumentointiin että ohjelma-ajojen tulosten esittämiseen. Tämän luvun eräiden esimerkkien ajossa oletamme, että kirjoitin on asianmukaisesti kytketty tietokoneeseen ja että siihen on tutustuttu ohjekirjan mukaisesti.

Ensimmäisenä esimerkkinä on näyte siitä, miten ohjelmalistaus aikaansaadaan kirjoittimen avulla. Kirjoita esimerkkiohjelma sivulta 58 näppäimistön avulla. Komennolla LIST PR: alkaa ohjelmalistauksen tulostus kirjoittimella (engl. PRinter).

```
LIST PR:
```

Jos käytössä on matriisikirjoitin, voi ohjelmalistaus näyttää seuraavalta:

```
10 PRINT CHR$(12)
20 PRINT "TIETOKONE ARPOO SATUNNAISLUKUJA"
30 PRINT "=============================="
40 PRINT
50 WHILE N<10
60   N=N+1
70   PRINT RND
80 WEND
90 PRINT
```

Kun tämä ohjelma ajetaan, ilmaantuu 10 satunnaislukua kuvaruudulle. Jos halutaan, että satunnaislukujen tulostus tapahtuu kirjoittimella, muutetaan ohjelmaa seuraavasti

```
4  REM OHJELMAN NIMI: ARVONTA
5  PREPARE "PR:" AS FILE 2
10  PRINT CHR$(12)
20  PRINT #2, "TIETOKONE ARPOO SATUNNAISLUKUJA"
30  PRINT #2, "====================="
40  PRINT #2,
50  WHILE N<10
60   N=N+1
70   PRINT #2,RND
80  WEND
90  PRINT
100 CLOSE 2
```

Tietokonejärjestelmässä käsitellään kirjoitinta ikään kuin se olisi 'peräkkäistiedosto' varustettuna varatulla nimellä PR: Tulostus näyttää seuraavalta:

```
TIETOKONE ARPOO SATUNNAISLUKUJA
===============================

 7.91124E-02
 4.41713E-02
 .765265
 .977572
 .207974
 .362484
 7.37993E-02
 .616114
 .210889
 .88768
```

Koska useimmat kirjoittimet syöttävät paperia vain yhdessä suunnassa, ei erityisfunktiota CUR(R,K) ole mahdollista käyttää kirjoituspään ohjaamiseen. Sijoittelu tiettyyn paikkaan rivillä voi sen sijaan tapahtua erityisfunktion TAB(T) avulla seuraavan ohjelmarivin osoittamalla tavalla:

```
70 PRINT #2, TAB(7) RND
```

Aja ohjelma tällä muutoksella ja huomioi lopputulos.

Mitä on sanojenkäsittely?

Asiakirjojen esittämisen helpottamiseksi käytetään nykyisin usein ns. mikrotietokonepohjaisia tietojenkäsittelyjärjestelmiä eli ns. sanastimia. Sopivan ohjelmiston ja hyvänlaatuisen kirjoittimen avulla voidaan useimpia yleisiä tietokonejärjestelmiä käyttää sanastimina. Täydellinen sanojenkäsittelyohjelma (tekstinkäsittelyohjelma) antaa käyttäjälle monta erilaista mahdollisuutta muokata kirjoitusta ja siitä tulee sen vuoksi hyvin laajakäyttöinen ohjelma. Kuvailemme muutamilla lyhyillä esimerkeillä eräitä BASIC-funktioita, jotka on tarkoitettu merkkijonojen (= merkeistä koostuvia yhdistelmiä) käsittelyyn.

Merkkijonon pituus

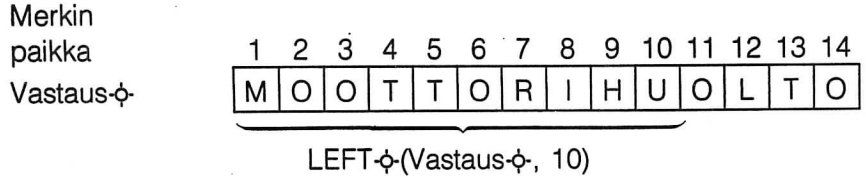
On usein välttämätöntä tietää, kuinka monta merkkiä on tietyssä merkkijonossa. Merkkijonon pituus (engl. LENght) määrätään funktion LEN(...) avulla

```
100 REM OHJELMAN NIMI: PITUUS
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "ANNA MIELIVALTAINEN MERKKIJONO"
140 INPUT " "Vastaus$
150 PRINT
160 Pituus = LEN(Vastaus$)
170 PRINT "ANNETUSSA MERKKIJONOSSA ON " Pituus " MERKKIÄ"
180 PRINT
190 END
```

Kun ohjelmarivi 160 suoritetaan, saa muuttuja Pituus arvokseen kokonaisluvun, joka ilmoittaa annetun merkkijonon merkkien lukumäärän (merkkijono annettiin rivillä 140 muuttujan Vastaus$ arvoksi).

Merkkijonomuuttujien käyttö syötön tarkkailussa

Koska käytettävissä oleva muisti on rajoitettu, on usein välttämätöntä rajoittaa syötettävien merkkijonojen pituutta. Funktion LEFT$(..., ...) avulla voidaan muodostaa merkkijono, joka koostuu annetun merkkijonon alkuosasta.



Tietokonejärjestelmässä käsitellään kirjoitinta ikään kuin se olisi 'peräkkäistiedosto' varustettuna varatulla nimellä PR: Tulostus näyttää seuraavalta:

```
TIETOKONE ARPOO SATUNNAISLUKUJA
===============================

 7.91124E-02
 4.41713E-02
 .765265
 .977572
 .207974
 .362484
 7.37993E-02
 .616114
 .210889
 .88768
```

Koska useimmat kirjoittimet syöttävät paperia vain yhdessä suunnassa, ei erityisfunktiota CUR(R,K) ole mahdollista käyttää kirjoituspään ohjaamiseen. Sijoittelu tiettyyn paikkaan rivillä voi sen sijaan tapahtua erityisfunktion TAB(T) avulla seuraavan ohjelmarivin osoittamalla tavalla:

```
70 PRINT #2, TAB(7) RND
```

Aja ohjelma tällä muutoksella ja huomioi lopputulos.

Mitä on sanojenkäsittely?

Asiakirjojen esittämisen helpottamiseksi käytetään nykyisin usein ns. mikrotietokonepohjaisia tietojenkäsittelyjärjestelmiä eli ns. sanastimia. Sopivan ohjelmiston ja hyvänlaatuisen kirjoittimen avulla voidaan useimpia yleisiä tietokonejärjestelmiä käyttää sanastimina. Täydellinen sanojenkäsittelyohjelma (tekstinkäsittelyohjelma) antaa käyttäjälle monta erilaista mahdollisuutta muokata kirjoitusta ja siitä tulee sen vuoksi hyvin laajakäyttöinen ohjelma. Kuvailemme muutamilla lyhyillä esimerkeillä eräitä BASIC-funktioita, jotka on tarkoitettu merkkijonojen (= merkeistä koostuvia yhdistelmiä) käsittelyyn.

Merkkijonon pituus

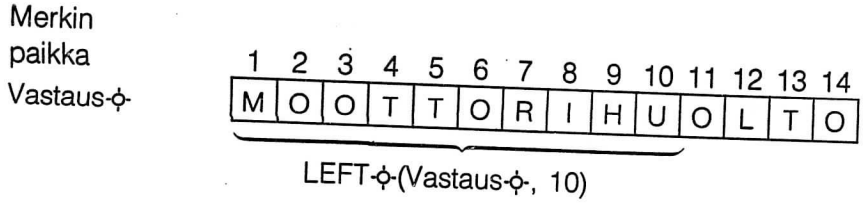
On usein välttämätöntä tietää, kuinka monta merkkiä on tietyssä merkkijonossa. Merkkijonon pituus (engl. LENght) määrätään funktion LEN(...) avulla

```
100 REM OHJELMAN NIMI: PITUUS
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "ANNA MIELIVALTAINEN MERKKIJONO"
140 INPUT " "Vastaus$
150 PRINT
160 Pituus = LEN(Vastaus$)
170 PRINT "ANNETUSSA MERKKIJONOSSA ON " Pituus " MERKKIÄ"
180 PRINT
190 END
```

Kun ohjelmarivi 160 suoritetaan, saa muuttuja Pituus arvokseen kokonaisluvun, joka ilmoittaa annetun merkkijonon merkkien lukumäärän (merkkijono annettiin rivillä 140 muuttujan Vastaus$ arvoksi).

Merkkijonomuuttujien käyttö syötön tarkkailussa

Koska käytettävissä oleva muisti on rajoitettu, on usein välttämätöntä rajoittaa syötettävien merkkijonojen pituutta. Funktion LEFT$(..., ...) avulla voidaan muodostaa merkkijono, joka koostuu annetun merkkijonon alkuosasta.



Seuraavassa muutamia esimerkkejä tämän funktion käytöstä:

```
100 REM OHJELMAN NIMI: ALKUOSA1
110 EXTEND
120 PRINT CHR$(12)
130 PRINT CUR(10,0) "ANNA MIELIVALTAINEN MERKKIJONO"
140 INPUT " "Test$
150 PRINT
160 PRINT CUR(5,0) "RAJOITETTU MERKKIJONO (MAX 10 MERKKIÄ)"
170 IF LEN(Test$)>10 THEN Test$=LEFT$(Test$,10)
180 PRINT Test$
190 PRINT CUR(20,0);
200 INPUT "TAHDOTKO JATKAA? (KYLLÄ/EI) "Vastaus$
210 IF LEFT$(Vastaus$,1)="K" OR LEFT$(Vastaus$,1)="k" THEN GOTO 130
220 PRINT
230 END
```

Rivillä 170 tarkistetaan muuttujan Test$ senhetkisen arvon pituus. Jos siinä on enemmän kuin 10 merkkiä, tulee muuttujan Test$ uudeksi arvoksi edellisen arvon 10 ensimmäistä merkkiä.

Rivillä 210 käytetään funktiota LEFT$ rajoittamaan vastaus ensimmäiseksi syötetyksi kirjaimeksi. Ehtolause rivillä 210 on totta, jos vastauksen ensimmäinen kirjain on K tai k.

Samaan tapaan voidaan merkkijonon loppuosa ja kappale välistä valita merkkijonofunktioilla RIGHT$(...,...) ja Mid$(...,...,...). Ks. yhteenveto.

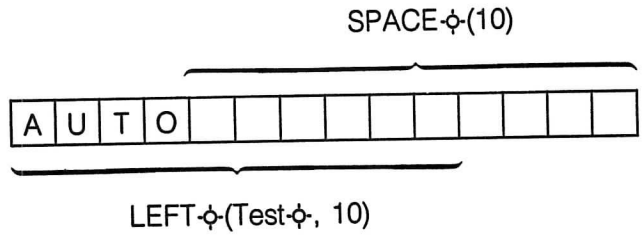
Merkkijonojen yhteenliittäminen

Mitä tapahtuu, jos edellisen ohjelman ajossa ensin syötät merkkijonon MOOTTORIHUOLTO ja sen jälkeen merkkijonon AUTO? Kuten näet, ei edellinen merkkijono tuhoudu kokonaan, kun lyhyempi kirjoitetaan samalle paikalle. Sama ilmiö tapahtuu, kun merkkijonoja tallennetaan suorasaantitiedostoon! Tämä ongelma voidaan ratkaista liittämällä sopiva määrä välilyöntejä lyhyemmän merkkijonon perään siten, että saadaan yhtä pitkiä merkkijonoja.

euraavasta kuviosta ilmenee mitä tapahtuu, kun ohjelmarivit 165
a 170 suoritetaan.

```
100 REM OHJELMAN NIMI: ALKUOSA2
110 EXTEND
115 DIM Test-◊-=170
120 PRINT CHR-◊-(12)
130 PRINT CUR(10,0) "ANNA MIELIVALTAINEN MERKKIJONO"
135 PRINT CUR(11,0) SPACE-◊-(160)
136 PRINT CUR(11,0);
140 INPUT " "Test-◊-
150 PRINT
160 PRINT CUR(5,0) "RAJOITETTU MERKKIJONO (MAX 10 MERKKIÄ)"
165 Test-◊-=Test-◊-+SPACE-◊-(10)
170 Test-◊-=LEFT-◊-(Test-◊-,10)
180 PRINT Test-◊-
190 PRINT CUR(20,0);
200 INPUT "HALUATKO JATKAA? (KYLLÄ/EI) "Vastaus-◊-
205 Vastaus-◊-=Vastaus-◊-+SPACE-◊-(1)
210 IF LEFT-◊-(Vastaus-◊-,1)="K" OR LEFT-◊-(Vastaus-◊-,1)="k" THEN GOTO 130
220 PRINT
230 END
```

Funktion SPACE-◊-(...) avulla voidaan muodostaa merkkijono,
ossa on haluttu määrä välilyöntejä.



Ohjelmarivin 135 tehtävänä on poistaa kuvaruudulta varmasti ko-
ko edellinen syöttö.

Jos rivillä 200 olevassa syötössä käytetään ainoastaan näppäintä
RETURN, tulee muuttujan Vastaus-◊- arvoksi tyhjä merkkijono.

Tyhjästä merkkijonosta puuttuu tietenkin myös ensimmäinen merkki. Tämän vuoksi aiheutuisi virhe, jos funktion LEFT-ϕ-(Vastaus-ϕ-, 1) arvo laskettaisiin. Jotta riviä 210 ajettaessa ei sattuisi ohjelmakatkosta, liitetään rivillä 205 merkkijonomuuttujan Vastaus-ϕ- arvon perään yksi välilyönti. Tällä tavoin voidaan yksi näppäimen RETURN painallus rinnastaa rivin 200 standardivastaukseen EI.

Äänimerkki kaiuttimesta

Kun tietokoneen käyttäjä syöttää merkkijonon, jota ei kokonaan voida hyväksyä, voidaan huomion herättämiseksi käyttää tietokoneen sisään rakennettua kaiutinta. Kun funktiota CHR-ϕ-(7) käytetään PRINT-lauseessa, se aikaansaa lyhyen äänimerkin. Lisäämällä seuraava ohjelmarivi saadaan varoitusmerkki, mikäli syötetty merkkijono on pitempi kuin 10 merkkiä.

```
145 IF LEN(Test-ϕ-)>10 THEN PRINT CHR-ϕ-(7)
```

Funktiossa CHR-ϕ-(...) oleva luku 7 on koodi, joka kertoo tietokoneelle, että kaiutin on otettava käyttöön. Jos käytetään koodia 12, on tuloksena kuvaruudun tyhjentyminen. Koodeja 65—94 käytetään vastaavalla tavalla aakkosten suurten kirjainten esittämiseen. Täydellistä koodia kutsutaan ASCII-koodiksi ja sitä käytetään useimmissa tietokoneissa kaikkien merkkien esittämiseksi. Vertaa ohjelmointikorttiin.

Merkkivalinta

Syötettyjen merkkijonojen tulkinta helpottuu usein, mikäli jonoissa on ainoastaan isoja kirjaimia. Jos ohjelmassa tapahtuu muutos pienistä kirjaimista suuriin, ei käyttäjälle tarvitse asettaa minkäänlaisia erikoisvaatimuksia.

ASCII-taulukosta ilmenee, että pienen kirjaimen koodi on aina 32 yksikköä suurempi kuin vastaavan suuren kirjaimen. Merkkijonon ensimmäisen merkin koodi annetaan funktiolla ASCII(...).

```
100 REM OHJELMAN NIMI: KIRJAIN
110 EXTEND
120 DIM Vastaus-ö- = 160
130 PRINT CHR-ö-(12)
140 PRINT "KIRJOITA RIVI SUURILLA JA PIENILLÄ KIRJAIMILLA"
150 PRINT
160 INPUT " "Vastaus-ö-
170 GOSUB 230 : REM KIRJAINVALINTA
180 PRINT CUR(8,0) "YHDENMUKAISTETTUNA:"
190 PRINT
200 PRINT Vastaus-ö-
210 PRINT
220 END
230 REM ------- VALINTA.SUB -------
240 FOR Paikka = 1 TO LEN(Vastaus-ö-)
250  Merkki-ö- = MID-ö-(Vastaus-ö-,Paikka,1)
260  Koodi = ASCII(Merkki-ö-)
270  IF Koodi > 95 THEN Koodi = Koodi—32
280  MID-ö-(Vastaus-ö-,Paikka,1) = CHR-ö-(Koodi)
290 NEXT Paikka
300 RETURN
```

Esimerkissä muutos pienistä kirjaimista suuriin tapahtuu aliohjelman VALINTA.SUB avulla. Silmukassa tutkitaan merkkijonoa Vastaus-ö- merkki merkiltä. Funktion MID-ö-(..., ..., ...) avulla valitaan merkki muuttujasta Vastaus-ö-. Kun rivi 260 suoritetaan, annetaan muuttujalle Koodi tutkitun merkin ASCII-arvo. Tarkistuksen ja mahdollisen koodin vaihdon jälkeen kirjoitetaan merkki takaisin oikealle paikalleen merkkijonoon Vastaus-ö-.

Yksinkertaista sanojenkäsittelyä

Tietokoneella tapahtuva sanojenkäsittely koostuu useista toiminnoista. Yleensä on mm. seuraavat mahdollisuudet:

- tekstissä olevan sanan tai kappaleen vaihto tai korjaus
- sanan automaattinen tavuihin jako
- *rivin ja sivun* pituuden muutos
- tekstin tulostus suorilla marginaaleilla
- tekstin tallennus ulkoiseen muistiin

BASIC:in merkkijonomuuttujien avulla voidaan nämä mahdollisuudet toteuttaa. Kirjoittakaamme esimerkiksi ohjelma, joka kautta koko tekstin vaihtaa sanan toiseksi samanpituiseksi sanaksi.

```
100 REM OHJELMAN NIMI: VAIHTO
110 EXTEND
120 PRINT CHR-ǫ-(12)
130 DIM Text-ǫ- = 1000
140 Osajono-ǫ- = "SUOMI"
150 PRINT "KIRJOITA RIVI, JOSSA SANA 'SUOMI' ESIINTYY MUUTAMIA
KERTOJA:"
160 PRINT
170 INPUT LINE Text-ǫ-
180 PRINT
190 PRINT
200 PRINT "UUSITTUNA:"
210 PRINT
220 Paikka = INSTR(1,Text-ǫ-,Osajono-ǫ-)
230 IF Paikka = 0 THEN GOTO 250 ELSE MID-ǫ-(Text-ǫ-,Paikka,5) = "NORJA"
240 GOTO 220
250 PRINT Text-ǫ-
260 END
```

Rivillä 170 oleva syöttö muuttujalle Teksti-ǫ- tapahtuu käskyn INPUT LINE ... avulla. Tämän käskyn etu verrattuna käskyyn INPUT ... on se, että kaikki merkit ovat hyväksyttyjä, myös pilkku ja lainausmerkit.

Kun riviä 220 suoritetaan, etsii tietokone osajonoa Osajono-ǫ- annetussa jonossa Teksti-ǫ-. Muuttuja Paikka saa arvokseen sen merkin paikan, missä Osajono-ǫ- ensimmäisen kerran esiintyy. Ellei Osajono-ǫ- esiinny jonossa Teksti-ǫ-, saa muuttuja Paikka arvon nolla.

Jos merkkijono "SUOMI" (Osajono-ǫ-) löytyy, muutetaan tämä osa merkkijonoksi "NORJA". Vaihtoa toistetaan, kunnes kaikki osajonot "SUOMI" on vaihdettu osajonoiksi "NORJA" koko tekstissä.

Voit kernaasti muuntaa ohjelmaa siten, että vaihdettavat merkkijonot voidaan valita mielivaltaisesti.

Lajittelu

Erilaisten luetteloiden tulostuksessa vaaditaan usein aakkosjärjestystä. Puhelinluettelo on hyvä esimerkki tällaisesta. Aakkosjärjestykseen lajittelu on mitä tavanomaisin tietokonetehtävä.

Voidakseen suorittaa lajittelua täytyy tietokoneen osata verrata merkkijonoja ja ratkaista kirjainjärjestys. Tämä tapahtuu tavanomaisilla operaattoreilla suurempi kuin (>), pienempi kuin (<) jne. Aakkosjärjestyksessä edellä olevat tulkitaan pienemmäksi kuin aakkosissa jäljempänä olevat merkkijonot. Vertailu tapahtuu siten, että kahta merkkijonoa verrataan merkki merkiltä, kunnes esiintyy kaksi eri merkkiä. Tietokone käyttää vastaavia ASCII-koodeja ratkaistessaan mikä merkki on pienin. Tästä seuraa valitettavasti, että merkkijonot, joissa on kirjain å, voivat tulla väärin sijoitetuksi.

Seuraavassa esitämme usein käytetyn menetelmän merkkijonojen vertailemiseksi ja muuttamiseksi niin, että lopuksi saadaan aakkosjärjestyksessä oleva luettelo. Tässä menetelmässä merkkijonoja verrataan pareittain ja ne jonot, jotka eivät ole aakkosjärjestyksessä vaihtavat paikkaa. Tässä ohjelmassa rajoitetaan merkkijonojen lukumäärä neljään. Jotta vertailujärjestys selviäisi, esitetään luettelo uudestaan jokaisen vertailun ja mahdollisen vaihdoksen jälkeen.

```
100 REM OHJELMAN NIMI: AAKKOSJ
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "NELJÄN NIMEN ASTEITTAINEN LAJITTELU"
140 PRINT "= = = = = = = = = = = = = = = = = = = = ="
150 PRINT
160 REM ------- SYÖTTÖ -------
170 REM
180 FOR Numero=1 TO 4
190  PRINT "ANNA NIMI " Numero;
200  INPUT " "Nimi$(Numero)
210 NEXT Numero
220 REM ------- LAJITTELU -------
230 REM
240 FOR Apaikka=1 TO 3
250  FOR Bpaikka=Apaikka+1 TO 4
260    IF Nimi$(Apaikka)<Nimi$(Bpaikka) THEN GOTO 300
270    Apumuuttuja$=Nimi$(Apaikka)
280    Nimi$(Apaikka)=Nimi$(Bpaikka)
290    Nimi$(Bpaikka)=Apumuuttuja$
300    GOSUB 390 : REM TULOSTUS
310  NEXT Bpaikka
```

(jatkuu)

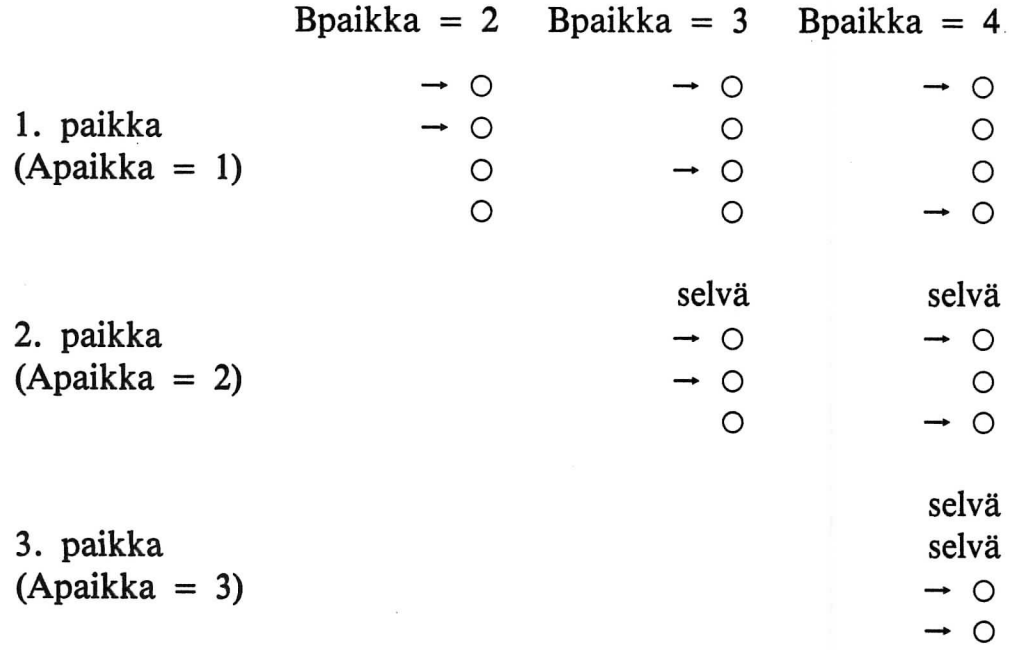
```
320 NEXT Apaikka
330 REM ------- LOPETUS -------
340 REM
350 PRINT : PRINT : PRINT
360 PRINT "LAJITTELU ON SUORITETTU JA AJO LOPPUNUT"
370 PRINT : PRINT
380 END
390 REM ------- TULOSTUS.SUB -------
400 PRINT
410 PRINT "VERTAILUN JA MAHD. PAIKANVAIHDON JÄLKEEN"
420 PRINT
430 FOR Numero=1 TO 4
440  IF Numero=Apaikka OR Numero=Bpaikka THEN PRINT "--> ";ELSE PRINT "   ";
450  PRINT Nimi$(Numero)
460  NEXT Numero
470 INPUT "      (Paina RETURN kun haluat jatkaa)"Odota$
480 RETURN
```

Kun käytetään merkkijonoja JENNY, JENNI, HENRIK ja HENRI, näyttää kuvaruutu kahden ensimmäisen vertailun jälkeen seuraavalta:

```
NELJÄN NIMEN ASTEITTAINEN LAJITTELU
===================================

ANNA NIMI  1  JENNY
ANNA NIMI  2  JENNI
ANNA NIMI  3  HENRIK
ANNA NIMI  4  HENRI

VERTAILUN JA MAHD. PAIKANVAIHDON JÄLKEEN

--> JENNI
--> JENNY
    HENRIK
    HENRI
       (Paina RETURN kun haluat jatkaa)

VERTAILUN JA MAHD. PAIKANVAIHDON JÄLKEEN

--> HENRIK
    JENNY
--> JENNI
    HENRI
       (Paina RETURN kun haluat jatkaa)
```

Kun ajat ohjelmaa, voit huomata, että 'pienin' merkkijono HENRI sijoittuu ensimmäiseen asemaan kolmen vertailun ja paikanvaihdoksen jälkeen. Kolme muuta merkkijonoa eivät vielä ole aakkosjärjestyksessä. Jotta löydettäisiin 'pienin' jäljelläolevista merkkijonoista JENNY, JENNI ja HENRIK, suoritetaan kaksi muuta vertailua ja mahdollista vaihdosta. Tämän jälkeen on HENRIK asettunut oikealle paikalleen asemaan 2. Jotta määräytyisi kumpi kahdesta merkkijonosta JENNY ja JENNI kuuluu kolmanneksi, tarvitaan vielä yksi vertailu.

Vertailut ja mahdolliset vaihdot oikeiden merkkijonojen sijoittamiseksi 1., 2. ja 3. paikalle ilmenevät seuraavasta kaaviosta:



Ohjelmajaksossa LAJITTELU tapahtuvat vertailut riviä 260 suoritettaessa ja mahdolliset paikanvaihdokset rivejä 270—290 suoritettaessa. Jottei mikään muuttujan arvo häviäisi, on käytettävä apumuuttujaa.

Tulostukseen tarvittavassa aliohjelmassa TULOSTUS.SUB on rivi 440 suorittamassa merkinnän niihin muuttujan arvoihin, joita ollaan vertaamassa.

Täydentämällä ohjelmaa LAJITTELU seuraavilla muutoksilla, voidaan suurempi määrä merkkijonoja lajitella aakkosjärjestykseen.

```
100 REM OHJELMAN NIMI: AAKKOSJ1
125 DIM Nimi$(24)
175 INPUT "LAJITELTAVIEN NIMIEN LUKUMÄÄRÄ "Määrä
176 PRINT
180 FOR Numero=1 TO Määrä
240  FOR Apaikka=1 TO Määrä—1
250    FOR Bpaikka=Apaikka+1 TO Määrä
430      FOR Numero=1 TO Määrä
```

Jos ajat ohjelman suurehkolla määrällä nimiä, saat käsityksen siitä valtavasta työmäärästä, joka tätä menetelmää käytettäessä tarvitaan luettelon järjestämiseksi aakkosjärjestykseen. Sen vuoksi vaativimpia luetteloita varten onkin käytettävissä tehokkaampia menetelmiä.

Yhteenveto

Käskyjä

INPUT LINE ...
Siirtää näppäimistöllä syötetyn rivin ilmoitettuun jonomuuttujaan. Kaikki merkit hyväksytään. Jonomuuttujan kaksi viimeistä merkkiä ovat aina CR (vaunupalautus) ja LF (rivinsiirto). Syötetyt merkit CR:ää ja LF:ää lukuunottamatta tulostetaan kuvaruutuun.

```
90 INPUT LINE S$
```

Merkkijonofunktioita

ASC(A$) · Vaihtoehto ASCII(A$)
Antaa ilmoitetun jonon ensimmäisen merkin ASCII-arvon.

```
20 LET P = ASC(B$)
```

CHR$(...)
Antaa merkin, joka vastaa ilmoitettua ASCII-arvoa.

```
10 PRINT CHR$(12)
20 P$ = CHR$(P, R, S, T, U, V)
```

INSTR(S, A$, B$)
Etsii osajonoa B$ jonossa A$ alkaen kohdasta S. Funktio antaa ensimmäisen sellaisen paikan, jossa B$ ensi kertaa esiintyy jonossa A$. Jos B$ ei esiinny A$:ssa, saadaan arvo 0.

```
30 R = INSTR(S, A$, B$)
```

LEFT$(A$, S)
Antaa merkkijonon A$ S ensimmäistä merkkiä.

```
30 D$ = LEFT$(C$, S)
```

LEN(A-ϕ-)
Antaa kokonaisluvun, joka vastaa ilmoitetun jonon pituutta.
20 X = LEN(A-ϕ-)

MID-ϕ-(A-ϕ-, S, T) Vaihtoehto MID(A-ϕ-, S, T)
Antaa T kpl merkkejä alkaen merkkipaikasta S ilmoitetussa merkkijonossa.
30 B-ϕ- = MID-ϕ-(A-ϕ-, S, T)

MID-ϕ-(A-ϕ-, S, T) = "..."
Vaihtaa T kpl merkkejä alkaen merkkipaikasta S jonossa A-ϕ- ilmoitettuun jonoon.
30 MID-ϕ-(A-ϕ-, 5, 4) = "UUSI"

RIGHT-ϕ-(A-ϕ-, S) Vaihtoehto RIGHT(A-ϕ-, S)
Antaa ilmoitetun jonon kaikki merkit alkaen kohdasta S.
40 B-ϕ- = RIGHT-ϕ-(A-ϕ-, S)

SPACE-ϕ-(S)
Antaa merkkijonon, jossa on S kpl välilyöntejä.
20 B-ϕ- = SPACE-ϕ-(253)

STRING-ϕ-(S, T)
Antaa jonon, jossa on S kpl merkkejä, joiden ASCII-arvo on T.
30 C-ϕ- = STRING-ϕ-(20, 45)

+
Yhdistää merkkijonoja.
30 S-ϕ- = A-ϕ- + B-ϕ- + C-ϕ- + ".BAS"

CHR-ϕ-(7)	Antaa piipauksen kaiuttimesta	(BELL)
CHR-ϕ-(10)	Rivinsiirto	(LF)
CHR-ϕ-(12)	Tyhjentää kuvaruudun ja siirtää kohdistimen vasempaan ylänurkkaan	(FF)
CHR-ϕ-(13)	Vastaa näppäimistön return-näppäintä	(CR)

Loogisia operaattoreita

OR
TAI.
On tosi, jos vähintään yksi operandeista on tosi.
30 IF A<B OR C = 10 THEN 40

Erityisfunktiot

TAB(T)
Siirtää kohdistimen rivin paikkaan T (T ⩾ 1).
Esiintyy ainoastaan PRINT-lauseissa.

```
40 PRINT TAB(14); "* * OTSIKKO * *"
```

Komento

LIST PR:
Tulostaa ABC800:ssa olevan ohjelman kirjoittimelle. PR:n vastine joissakin ohjelmissa on V24:

```
LIST PR:          (kaikki rivit)
LIST PR:, 10—90 (rivit kahden rivinumeron
                  välillä)
```

Harjoitustehtäviä

* 9.1 Seuraavan ohjelman avulla voidaan nimesi kirjoittaa 30 kertaa paperille. Muuta ohjelmaa siten, että nimi kirjoitetaan 30:een eri tabulaattoriasemaan.

```
100 REM OHJELMAN NIMI: TULOSTUS
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "TARKISTA ETTÄ KIRJOITIN ON KYTKETTY"
140 PRINT
150 INPUT "ANNA NIMESI:",Nimi$
160 PREPARE "PR:" AS FILE 1
170 FOR Kierros=1 TO 30
180 PRINT #1,TAB(10) Nimi$
190 NEXT Kierros
200 END
```

* 9.2 Muuta ohjelmaa LUE1 sivulla 128 siten, että tuotenumeron, tuotenimen ja lukumäärän tulostus tapahtuu kirjoittimella.

* 9.3 Täydennä ohjelmaa KIRJ5 sivulla 140 siten, että tuotenimikenttä saa 36 merkin vakiopituuden.

* 9.4 Vastaperustetussa kiinalaisravintolassa ANIHC oli tietokoneista kiinnostunut kokki. Hän ehdotti, että ruokalistan rivit alettaisiin kirjoittaa siten, että ne voisi lukea oikealta vasemmalle. Auta kokkia ratkaisemaan ongelma! (KCUL DOOG)

* 9.5 Salainen poliisi käyttää oveluuksissaan seuraavaa ohjelmaa sanoman muuntamiseksi:

```
100 REM OHJELMAN NIMI: SUPER
110 PRINT CHR-ϕ-(12)
120 GET S-ϕ-
130 IF S-ϕ- = CHR-ϕ-(13) THEN GOTO 160
140 PRINT CHR-ϕ-(ASCII(S-ϕ-)+1);
150 GOTO 120
160 END
```

Agentti DNOB SEMAJ lähetti eräänä päivänä seuraavan viestin:

```
FGCTL 3
```

Kirjoita ohjelma, joka muuttaa viestin selväkieliseksi.

* 9.6 Sanojenkäsittelyssä voi eräänä toimintona olla tekstin alussa otsikon kirjoittaminen keskelle sivua ja sen automaattinen alleviivaaminen. Laadi ohjelmajakso, joka suorittaa tällaisen tehtävän.

* 9.7 Täydennä ohjelmaa KILPA2 sivulla 109 siten, että meno jaksoon VIRHEENKÄSITTELIJÄ aikaansaa varoitusäänen.

*9.8 Kirjoita ohjelma, joka lukee tiedoston VARASTO3.DAT ja esittää tuotenimikkeistä ja määristä aakkostetun luettelon.

* 9.9 Muunna ohjelmaesimerkkiä AAKKOSJ siten, että 100 nimeä voidaan lajitella aakkosjärjestykseen. Esimerkissä olevat vertailut ja vaihdot jätetään pois. Aakkostettu luettelo kirjoitetaan paperille.

* 9.10 Muunna sivun 58 ohjelmaa siten, että se toimii vaikka salainen koodi annettaisiin pienillä kirjaimilla. Muuta ohjelmaa myös siten, että äänimerkki varoittaa, mikäli otto ylittää senhetkisen saldon. Anna kuitille nimi PIKAOTTO.

10

Matemaattisia laskuja

Tarkkoja laskuja

Tietyissä tapauksissa täytyy lasku voida suorittaa useamman kuin seitsemän numeron tarkkuudella. Esim. loppusumman 165732,25 esittäminen vaatii kahdeksan (merkitsevän) numeron tarkkuutta. Tähän asti käytetty tarkkuus ei siis riitä. BASIC II:ssa voidaan numeroiden määrää lisätä kuuteentoista. Tämä tapahtuu sijoittamalla ohjelmassa ensimmäiseksi käsky DOUBLE.

Seuraavassa esimerkkiohjelmassa käytetään käskyä DOUBLE.

```
100 REM OHJELMAN NIMI: DOUBLE1
110 DOUBLE
120 EXTEND
130 PRINT CHR-ϕ-(12)
140 INPUT "ANNA LAINAN MÄÄRÄ (MK)",Laina
150 INPUT "ANNA KORKOPROSENTTI ",Korkoprosentti
160 PRINT
170 Korko=Laina*Korkoprosentti/100
180 PRINT CHR-ϕ-(12)
190 PRINT "LAINAN MÄÄRÄ (MK)",Laina
200 PRINT "KORKOPROSENTTI ",Korkoprosentti
210 PRINT
220 PRINT "KORKO VUODESSA (MK)",Korko
230 PRINT
240 END
```

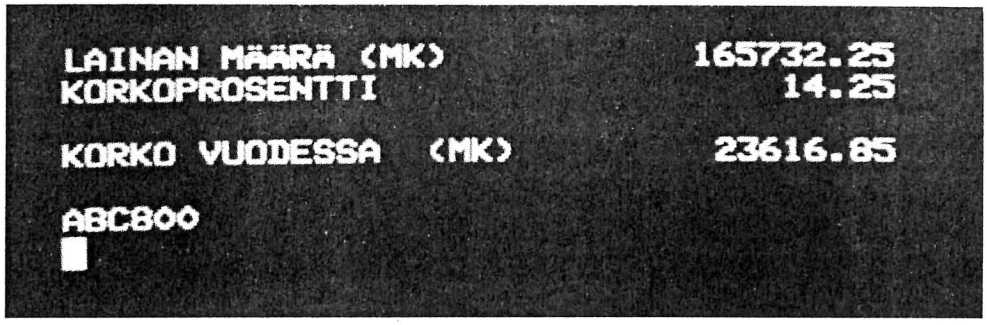
Tämä ohjelma toimii myös hyvin suurilla lukumäärillä. Oikean tulostuksen saamiseksi markoissa ja penneissä on tulostusta tietenkin muotoiltava tavalliseen tapaan käskyn PRINT USING "..."... avulla. Lisättävät ohjelmarivit ilmenevät seuraavasta:

```
100 REM OHJELMAN NIMI: DOUBLE2
110 DOUBLE
120 EXTEND
130 PRINT CHR-ϕ-(12)
140 INPUT "ANNA LAINAN MÄÄRÄ (MK)",Laina
150 INPUT "ANNA KORKOPROSENTTI ",Korkoprosentti
160 PRINT
170 Korko=Laina*Korkoprosentti/100
```

(jatkuu)

```
180 PRINT CHR$(12)
190 PRINT "LAINAN MÄÄRÄ (MK)",
195 PRINT USING "##########.##" Laina
200 PRINT "KORKOPROSENTTI ",
205 PRINT USING "##########.##" Korkoprosentti
210 PRINT
220 PRINT "KORKO VUODESSA (MK)",
225 PRINT USING "##########.##" Korko
230 PRINT
240 END
```

Ajon jälkeen voi kuvaruutu näyttää seuraavalta:



Numeroiden määrää voi vaihtoehtoisesti rajoittaa myös käskyllä DIGITS ... Katso tarkemmin yhteenvedosta ja harjoitustehtävistä.

Erilaisia muuttujatyyppejä

Ne muuttujalajit, joihin tähän mennessä olemme tutustuneet, ovat olleet merkkijonomuuttujia ja lukumuuttujia. Niitä lukumuuttujia, joita olemme käyttäneet, kutsutaan myös liukulukumuuttujiksi ja ne voivat esittää lukuja alueissa $\pm 1{,}7 \cdot 10^{-37}$ ... $\pm 1{,}7 \cdot 10^{38}$. Liukulukujen numeroiden määrä voidaan valita käskyillä DOUBLE ja SINGLE.

Tietyissä sovellutuksissa, joissa on tärkeää saada lyhyt suoritusaika tai säästää muistitilaa, käytetään kokonaislukumuuttujia liukulukumuuttujien sijasta. Lukualue kokonaislukumuuttujalle on rajoitettu -32768:sta $+32767$:ään. Numeroiden määrään ei voida tällöin vaikuttaa käskyillä SINGLE tai DOUBLE.

Erilaisten lukumuuttujien erottamiseksi toisistaan lopetetaan kokonaislukumuuttujien nimet %-merkkiin, esim. A%, Kierros% ja Tietue% (Indeksi%). Vakiot voidaan antaa samalla tavalla liukulukuina tai kokonaislukuina. Seuraavasta ohjelmaesimerkistä ilmenee, miten muuttujatyypin valinta vaikuttaa suoritusaikaan. Ajan mittaus sujuu tietokoneen sisäisellä kellolla, jonka tulee olla asetettu aikaan ennen ajoa. Vrt. sivu 60!

Tyhjennä tietokoneen muisti NEW-komennolla ennen seuraavan ohjelman kirjoittamista.

```
100 REM OHJELMAN NIMI: FLOATDOU
110 DOUBLE
120 EXTEND
130 PRINT CHR$(12)
140 PRINT "LIUKULUKU — DOUBLE"
150 PRINT "= = = = = = = = = = = = = ="
160 PRINT CUR(5,0) "LÄHTÖAIKA: " TIME$
170 PRINT CUR(20,0) "SUORITUS KÄYNNISSÄ, ODOTA !"
180 FOR Kierros=1 TO 10000
190 NEXT Kierros
200 PRINT CUR(7,0) "LOPPUAIKA: " TIME$
210 PRINT CUR(20,0),SPACE$(40)
220 END
```

Yllä olevassa ohjelmassa esitettiin liukulukumuuttuja Kierros kaksinkertaisella (engl. DOUBLE) tarkkuudella. Suoritusaika oli noin 12 sekuntia.

Paluu yksinkertaiseen (engl. SINGLE) esitykseen tapahtuu käskyllä SINGLE. Huom! Ennen ohjelmarivin 110 editointia on käytettävä komentoa CLEAR.

```
100 REM OHJELMAN NIMI: FLOATSIN
110 SINGLE
120 EXTEND
130 PRINT CHR$(12)
140 PRINT "LIUKULUKU — SINGLE"
150 PRINT "= = = = = = = = = = = ="
160 PRINT CUR(5,0) "LÄHTÖAIKA: " TIME$
170 PRINT CUR(20,0) "SUORITUS KÄYNNISSÄ, ODOTA !"
180 FOR Kierros=1 TO 10000
190 NEXT Kierros
200 PRINT CUR(7,0) "LOPPUAIKA: " TIME$
210 PRINT CUR(20,0) SPACE$(40)
220 END
```

Yksinkertaisen esityksen hyötynä on selvästi lyhyempi suoritusaika.

Dramaattisempi ajanvoitto saavutetaan muuttamalla liukulukumuuttuja Kierros kokonaislukumuuttujaksi Kierros%.

```
100 REM OHJELMAN NIMI: INTEGER
120 EXTEND
130 PRINT CHR$(12)
140 PRINT "KOKONAISLUKU"
150 PRINT "= = = = = = = = ="
160 PRINT CUR(5,0) "LÄHTÖAIKA: " TIME$
170 PRINT CUR(20,0) "SUORITUS KÄYNNISSÄ, ODOTA !"
180 FOR Kierros% = 1% TO 10000%
190 NEXT Kierros%
200 PRINT CUR(7,0) "LOPPUAIKA: " TIME$
210 PRINT CUR(20,0) SPACE$(40)
220 END
```

Kun ohjelman lukumuuttujat suurimmaksi osaksi ovat kokonaislukumuuttujia, voidaan syöttöä yksinkertaistaa sijoittamalla käsky INTEGER ohjelman alkuun. Tällöin ei kokonaislukumuuttujia tarvitse merkitä %-merkillä, mutta sitävastoin liukulukumuuttujat on merkittävä pisteellä, esim. A., Tulos. ja Tulos. (Vuosi). Käyttämällä tällaista käskyä ohjelman alussa tulee ohjelmateksti havainnollisemmaksi, mikäli ohjelma sisältää useita kokonaislukumuuttujia. Käskyllä FLOAT voidaan tietokoneen toiminta palauttaa normaaliin muuttujamerkintöjen tulkintaan. Tämä merkitsee, että ainoastaan kokonaislukumuuttujat merkitään.

ASCII-aritmetiikkaa

Mikäli jossakin tapauksessa 16 numeroa ei riittäisi, voidaan käyttää erityistä ASCII-aritmetiikkaa, jossa käsitellään lukuja aina 125 numeron tarkkuuteen asti. Tässä tapauksessa käytetään ns. numeerisia merkkijonoja, ts. lukuja tallennettuina merkkijonojen muotoon. Ks. tarkemmin yhteenvedosta ja harjoitustehtävistä.

Aritmeettisia lausekkeita

Olemme useissa ohjelmaesimerkeissä jo käyttäneet yksinkertaisia aritmeettisia lausekkeita. Mutkikkaammissa lausekkeissa on tärkeätä ottaa huomioon matemaattisten operaatioiden sisäinen arvojärjestys. Tämä järjestys selviää ohjelmointikortista.

Huomaa että suluissa oleva lauseke aina lasketaan ensisijaiseksi. Seuraavassa esimerkkejä aritmeettisten lausekkeiden kirjoittamisesta BASIC-lauseiksi:

Tavallinen	**BASIC**
$kx + 1$	K*X + L
$K(1 + \frac{P}{100})^T$	K*(1 + P/100)**T
$a^2 + b^2$	A**2 + B**2
$\frac{a + b}{c + d}$	(A + B)/(C + D)

Matemaattiset perusfunktiot, esim. trigonometriset funktiot ja logaritmit, ovat tietenkin käytettävissä myös BASIC:issa. Seuraavassa joitakin esimerkkejä siitä, kuinka matemaattisia perusfunktioita voidaan käyttää BASIC:issa. Kaikki funktiot esitellään yhteenvedossa.

Tavallinen	**BASIC**
$\sqrt{x}$	SQR(X)
$\frac{1 + \cos(2x)}{2}$	(1 + COS(2*X))/2
$E(1 - e^{-t/RC})$	E*(1 − EXP(−T/(R*C)))
$\sin x \cdot e^{-0,3x}$	SIN(X)*EXP(−.3*X)

Kotkan
kaupunginkirjasto

Tuloksen esittäminen

Seuraavassa esimerkki matemaattisten funktioiden käytöstä. Funktiolle Y(X), joka esittää 'vaimenevaa aaltoa', lasketaan joukko arvoja. Ajossa voi valita taulukon esittämisen joko paperilla tai kuvaruudulla.

```
100 REM OHJELMAN NIMI: FUNKTAUL
110 DOUBLE
120 EXTEND
130 PRINT CHR$(12)
140 INPUT "TOIVOTKO PAPERITULOSTUSTA? (KYLLÄ/EI) "Vastaus$
150 IF Vastaus$ < > "EI" THEN F=FNPaperi ELSE F=0
160 PRINT #F,
170 PRINT #F, "TAULUKKO FUNKTIOLLE Y=SIN(X)*EXP(—.3*X)"
180 PRINT #F, "= = = = = = = = = = = = = = = = = = = = = = = = =.= ="
190 PRINT #F,
200 PRINT #F,TAB(5) "X" TAB(15) "Y=SIN(X)*EXP(—.3*X)"
210 PRINT #F,
220 FOR X=0 TO 2*PI STEP PI/6
230  Y=SIN(X)*EXP(—.3*X)
240  PRINT #F, USING "# #.# # # # # # #" X TAB(20) Y
250 NEXT X
260 PRINT
270 PRINT "AJO SUORITETTU"
280 END
290 REM ------- PAPERI.DEF -------
300 DEF FNPaperi
310  OPEN "PR:" AS FILE 1
320  PRINT "ODOTA!"
330  RETURN 1
340 FNEND
```

Kun ohjelmarivi 150 suoritetaan, annetaan muuttujalle F arvo 1, jos tulostus tapahtuu kirjoittimella ja arvo 0, jos tulostus tapahtuu kuvaruudulla. Tietokonejärjestelmä pitää kuvaruutua aina avoimena tiedostona, jonka tiedostonumero on 0.

Arvojen tallennus myöhempää tarvetta varten

Ohjelmiston käyttö esim. graafiseen esitykseen edellyttää usein, että numeroaineisto on tallennettuna tiedostoon. Tällaisia tiedostoja voidaan muodostaa ajamalla laskentaohjelma tai tiedonkeruuohjelma.

Tällaisen tiedoston aikaansaamiseksi (mm. graafista esitystä varten) voidaan edellistä ohjelmaa muuttaa siten, että muodostetaan 100 lukuparia käsittävä tiedosto.

```
100 REM OHJELMAN NIMI: FUNKTIED
110 DOUBLE
120 PRINT CHR$(12)
130 PRINT "FUNKTIOLLE Y=SIN(X)*EXP(—3*X) LASKETAAN"
140 PRINT "SATA ARVOA,   0 < = X < = 4*PI"
150 PRINT "LUKUPARI (X,Y) TALLENNETAAN TIEDOSTOON"
160 PRINT "'LUKUPARI.DAT'"
170 PRINT
180 PRINT "ODOTA!"
190 PREPARE "LUKUPARI.DAT" AS FILE 1
200 FOR X=0 TO 4*PI STEP PI/25
210  PRINT #1,X
220  Y=SIN(X)*EXP(—.3*X)
230  PRINT #1,Y
240 NEXT X
250 CLOSE 1
260 PRINT
270 PRINT "TIEDOSTO VALMIS"
280 END
```

Oman matemaattisen funktion määrittely

BASIC:issa olevien standardifunktioiden lisäksi voidaan määritellä myös omia funktioita käskyn DEF FN ... (...) = ... avulla. Vaikka funktioiden määrittely voidaan sijoittaa mielivaltaiseen paikkaan, laitetaan se mieluiten ohjelman loppuun.

Seuraavassa ohjelmaesimerkissä määritellään erityinen funktio nesteen tilavuuden laskemiseksi pallonmuotoisessa tankissa. Sisäl-

lön tilavuus lasketaan tankin säteen (R) ja tankin yläosan ja nestepinnan etäisyyden (H) funktiona seuraavan kaavan mukaan:

$$T(R,H) = \frac{\pi}{3}(2R - H)^2 (R + H)$$

```
100 REM OHJELMAN NIMI: TILAVUUS
110 DOUBLE
120 EXTEND
130 PRINT CHR-ö-(12)
140 PRINT "PALLON MUOTOISEN TANKIN SISÄLLÖN TILAVUUS"
150 PRINT "= = = = = = = = = = = = = = = = = = = = = = = = = = = = = ="
160 PRINT
170 INPUT "ANNA TANKIN SÄDE (m) ",Säde
180 PRINT
190 PRINT "ETÄISYYS TANKIN HUIPULTA"
200 INPUT "NESTEEN PINTAAN (M) ",Korkeus
210 IF Korkeus>2*Säde THEN Korkeus=2*Säde
220 PRINT
230 PRINT "TANKIN SISÄLTÖ (m3) ON ";
240 PRINT USING "# # # # # #.# # #" FNTilavuus(Säde,Korkeus)
250 PRINT
260 PRINT "TILAVUUS PROSENTEISSA KOKO TANKIN TILAVUUDESTA ";
270 PRINT USING "# # #" FNTilavuus(Säde,Korkeus)/FNTilavuus(Säde,0)*100
280 PRINT
290 PRINT "AJO SUORITETTU"
300 END
310 REM ------- TILAVUUS.DEF -----------
320 DEF FNTilavuus(Säde,Korkeus)=1/3*PI*(2*Säde—Korkeus)ü2*(Säde+Korkeus)
```

Ohjelmajakso funktiona

Funktiota ei aina tarvitse määritellä ohjelmariville muotoiltuna aritmeettisena lausekkeena. Uudenaikaisten BASIC-muotojen etuna on se, että funktio voidaan määritellä ohjelmajaksona, joka sisältää useita ohjelmarivejä. Vaikka ohjelmajaksossa suoritetaan monia tehtäviä, täytyy funktion arvoksi aina tulla luku tai merkkijono.

Funktion FNMaxarvo (Tiedosto-ö-) arvoksi tulee muuttujan Ymax arvo, kun rivi 330 suoritetaan. Funktion saama arvo riippuu siitä, mikä tiedosto (Tiedosto-ö-) annetaan.

```
100 REM OHJELMAN NIMI: MAXARVO
110 EXTEND
120 PRINT CHR-ö-(12)
130 PRINT "ARVOJEN Y SUURIMMAN ARVON MÄÄRÄÄMINEN,"
140 PRINT "KUN LUKUPARIT (X,Y) OVAT TALLENNETTUINA"
150 PRINT "ANNETUSSA PERÄKKÄISTIEDOSTOSSA"
160 PRINT
170 PRINT "ANNA TOIVOTUN TIEDOSTON NIMI"
180 INPUT "(ESIM. 'LUKUPARI.DAT')",Tiedosto-ö-
190 PRINT
200 PRINT "MAXIMIARVO ON " FNMax(Tiedosto-ö-)
210 PRINT
220 END
230 REM ------- MAXARVO.DEF -------
240 DEF FNMax(Tiedosto-ö-) LOCAL Xtest,Ytest,Ymax
250  Ymax=0
260  OPEN Tiedosto-ö- AS FILE 1
270  ON ERROR GOTO 340
280  INPUT #1,Xtest
290  INPUT #1,Ytest
300  IF YTest>Ymax THEN Ymax=Ytest
310  GOTO 280
320  CLOSE 1
330  RETURN YMax
340  RESUME 320
350 FNEND
```

isäys LOCAL Xtest,Ytest,Ymax rivillä 240 merkitsee, että anne-
ıt muuttujat ovat paikallisia, ts. samoja muuttujien nimiä voi-
aan käyttää funktion määrittelyjakson ulkopuolella vaikuttamat-
ı näihin muuttujanarvoihin. Tätä mahdollisuutta ei käytetä tässä
hjelmassa, mutta määrittelemällä paikallisia muuttujia helpote-
ıan saman funktiomäärittelyn käyttöä eri ohjelmissa.

Vakioita ohjelmassa

ietokoneilla tapahtuvassa tietojenkäsittelyssä muutetaan yleensä
etoja ajosta toiseen. Tällöin tiedot tallennetaan ulkopuoliseen
ıuistiin kuten tiedostoihin. Tietyissä tapauksissa, kun samaa tie-
ba käytetään muuttamattomana ajosta toiseen, voi tämä tieto

muodostaa osan ohjelmatekstistä. Tieto tallennetaan silloin käskyn DATA jälkeen pilkuilla erotettuna samalla tavalla kuin peräkkäistiedostossa. DATA-lauseet voidaan sijoittaa mielivaltaiseen paikkaan ohjelmassa, mutta lauseiden keskinäinen järjestys on tietenkin olennainen.

Tiedon luku DATA-lauseista tapahtuu käskyllä READ ... alkaen ensimmäisestä ohjelman datasta, ts. vastaavalla tavalla kuin luku peräkkäistiedostosta. Käskyn RESTORE ... avulla alkaa luku annetusta DATA-lauseesta lähtien.

Seuraavassa ohjelmassa ovat kuukauden nimet sijoitettuina DATA-lauseisiin:

```
100 REM OHJELMAN NIMI: DATA
110 EXTEND
120 PRINT CHR$(12)
130 DIM Kuukausi$(12)
140 REM
150 DATA TAMMIKUU,HELMIKUU,MAALISKUU,HUHTIKUU,TOUKOKUU,KESÄKUU
160 DATA HEINÄKUU,ELOKUU,SYYSKUU,LOKAKUU,MARRASKUU,JOULUKUU
170 REM
180 FOR Numero=1 TO 12
190  READ Kuukausi$(Numero)
200 NEXT Numero
210 Kuukaudennumero$=MID$(TIME$,6,2)
220 Kuukaudennumero=VAL(Kuukaudennumero$)
230 PRINT "KUUKAUSITIEDOTE: " Kuukausi$(Kuukaudennumero)
240 END
```

Riveillä 180—200 olevan silmukan avulla annetaan muuttujalle Kuukausi$(1) arvo TAMMIKUU, muuttujalle Kuukausi$(2) arvo HELMIKUU jne. Mikäli sisäänrakennettu kello on asetettu oikeaan aikaan, löytyy kuluvan kuun numero merkkijonosta TIME$ paikoilta 6 ja 7. Nämä kaksi merkkiä viedään muuttujan Kuukaudennumero$ arvoksi, kun rivi 210 suoritetaan. Merkkijonofunktion VAL(...) avulla voidaan merkkijono muuttaa lukumuuttujaksi ja tämä tapahtuu, kun rivi 220 suoritetaan. Tämän jälkeen kirjoitetaan kuukauden nimi selväkielisenä riviä 230 suoritettaessa.

Käskyjä

DATA ...

Tallentaa muuttuja-arvot, jotka luetaan sitten READ-lauseella. Merkkijonot, jotka sisältävät pilkkuja, tulee ympäröidä lainausmerkeillä.

```
40 DATA 13, 2, 6, RIITTA
50 DATA "Allekirjoittanut, PELLE PELKONEN"
```

DEF FN ... (...) = ...

Määrittelee oman funktion yhdellä rivillä.

```
70 DEF FNY(R) = PI*R*R
80 DEF FNZ(B, H) = (B*H)/2
90 DEF FNX$(S$) = LEFT$(S$,LEN(A$) - 2)
```

DEF FN ... (...) LOCAL ...

Määrittelee oman monirivisen funktion. Ilmoituksella LOCAL ... määritellään paikallismuuttujat. LOCAL ... voidaan jättää pois. Funktion arvo palautetaan käskyllä RETURN ... FNEND-käsky lopettaa määrittelyn.

```
100 DEF FNY(R) LOCAL(S)
200 S = R/180
300 RETURN S*R
400 FNEND
```

DIGITS ...

Rajoittaa esitettävien numeroiden määrän tulostettaessa muuttujan arvoja käskyllä PRINT ... Ei vaikuta laskentatarkkuuteen.

```
10 DIGITS 4
70 DIGITS N
```

DOUBLE

Aiheuttaa liukulukujen esittämisen 16 numeron tarkkuudella normaalin 7 numeron sijasta. Käskyn tulee olla ensimmäisenä ohjelmassa. Ks. myös SINGLE.

FLOAT

Aiheuttaa sen, että muuttuja-arvoja käsitellään normaalisti liukulukuina. Kokonaislukumuuttujat ja kokonaislukuvakiot merkitään %-merkillä. FLOAT sijoitetaan ensimmäiseksi ohjelmaan. Vrt. INTEGER.

```
10 FLOAT
20 A = 2.567 : B% = 9%
```

FNEND
Lopettaa oman monirivisen funktion määrittelyn. Ks. DEF FN ... (...)

INTEGER
Aiheuttaa sen, että muuttuja-arvoja ja vakioita käsitellään normaalisti kokonaislukuina. Liukulukumuuttujat ja -vakiot merkitään pisteillä. INTEGER sijoitetaan ohjelmaan ensimmäiseksi. Vrt. FLOAT.

```
10 INTEGER
20 A. = 2.567 : B = 9
```

READ ...
Sijoittaa muuttujalle arvot DATA-lauseesta.

```
10 READ B, C-o-
```

RESTORE ...
Aiheuttaa sen, että seuraava READ-lause lukee dataa alkaen tietystä DATA-lauseesta.

```
50 RESTORE
```

(Lukee dataa alkaen ensimmäisestä Data-lauseesta.)

```
70 RESTORE 100
```

(Lukee dataa DATA-lauseesta alkaen riviltä 100.)

RETURN
Aiheuttaa paluuhypyn alirutiinista vastaavaa GOSUB-lausetta seuraavaan lauseeseen. Vrt. DEF FN ... (...)

SINGLE
Aiheuttaa sen, että liukuluvut esitetään 7 numerolla. Kumoaa DOUBLE-käskyn vaikutuksen.

Merkkijonofunktioita

NUM-o- (S)
Antaa numeerisen merkkijonon, jossa on samat merkit kuin ne, jotka ilmoitettu lukumuuttuja tulostaa PRINT-lauseella. Blanko, joka on varattu plus-merkkiä varten, ei sisälly jonoon NUM-o-(S).

```
10 B-o- = NUM-o-(S)
```

VAL(A-o-)
Muuttaa ilmoitetun numeerisen jonon lukumuuttujaksi.

```
10 S = VAL(A-o-)
20 S = VAL(".000000003")
```

ASCII-funktiot

ADD$-(A$-, B$-, T)
Summaa ilmoitetut numeeriset merkkijonot. Tulos pyöristetään ilmoitettuun (T) määrään oikeita desimaaleja. Numeeriset merkkijonot saavat sisältää numeroita (enintään 124 kpl), merkit + ja − sekä desimaalipisteen.

```
20 C$ = ADD$(A$, B$, T)
```

DIV$(A$, B$, T)
Jakaa numeerisen merkkijonon (A$) toisella (B$).
Vrt. ADD$(A$, B$, T)

```
30 C$ = DIV$(A$, B$, T)
```

MUL$(A$, B$, T)
Kertoo numeerisen merkkijonon (A$) toisella (B$).
Vrt. ADD$(A$, B$, T).

```
40 D$ = MUL$(A$, B$, T)
```

SUB$(A$, B$, T)
Vähentää numeerisen merkkijonon (A$) toisesta (B$).
Vrt. ADD$(A$, B$, T).

```
50 C$ = SUB$(A$, B$, T)
```

COMP%(A$, B$)
Vertaa kahta numeerista jonoa. Funktio antaa seuraavat arvot vastaukseksi:

```
−1, jos A$ < B$
 0, jos A$ = B$
+1, jos A$ > B$

10 P = COMP%(A$, B$)
```

Matemaattiset funktiot

ABS (X)	x:n itseisarvo, $\|x\|$.
ATN (X)	x:n arctangentti.
COS (X)	x:n kosini, x radiaaneina.
EXP (X)	e^x.
FIX (X)	x:n kokonaisosa.
INT (X)	suurin kokonaisluku, joka on pienempi tai yhtä suuri kuin x, [x].
LOG (X)	e-kantainen logaritmi eli luonnollinen logaritmi.
LOG10 (X)	10-kantainen logaritmi.
MOD (X, Y)	jakolaskun X/Y jakojäännös.
SGN (X)	-1, jos $x < 0$ 0, jos $x = 0$ 1, jos $x > 0$.
SIN (X)	x:n sini, x radiaaneina.
SQR (X)	x:n neliöjuuri, $\sqrt{x}$.
TAN (X)	x:n tangentti, x radiaaneina.
PI	π, luku 3.14159 (3.141592653589793).
RND	esittää satunnaisluvun väliltä 0—0.999999 (vaihtoehtoisesti väliltä 0—0.9999999999999999).

Harjoitustehtäviä

* 10.1 Kirjoita ohjelma, joka esittää taulukon, jossa on noin 15 kpl funktion $y = \sin(x)$ arvoja väliltä $0 \leqslant x \leqslant 2\pi$. Anna arvot 10 desimaalin tarkkuudella.

* 10.2 Seuraava valokuva esittää luvun π erilaisilla tarkkuuksilla. Laadi ohjelma, joka antaa tällaisen tulostuksen.

* 10.3 Tietokoneet käyttävät binääristä lukujärjestelmää. Tästä syystä 2:n potenssit esiintyvät usein tietojenkäsittelyssä. Laske sopivalla tarkkuudella kaikki 2:n potenssit 2:sta 2^{50}:een.

* 10.4 Pääoma K_0 kasvaa korkoa korolle kaavan

$$K = K_0 (1 + \frac{P}{100})^T$$

mukaan, missä P on korkoprosentti ja T aika vuosissa. Kirjoita ohjelma, missä tätä kaavaa käytetään funktion määrittelemiseen yhdellä ohjelmarivillä.

10.5 Laske cos(x) sadalle x:n arvolla väliltä $0 \leqslant x \leqslant 2\pi$. Tallenna yhteenkuuluvat arvot x ja cos(x) peräkkäistiedostoksi.

10.6 Käytä ohjelmaa MAXARVO ottaaksesi esiin suurimman arvon cos(x), joka on tallennettuna edellisessä tehtävässä muodostettuun tiedostoon. Muuta sitten ohjelma sellaiseksi, että se esittää pienimmän arvon.

* 10.7 Laadi ohjelma, joka laskee keskiarvon 15 luvulle, jotka annetaan näppäimistön avulla. Ohjelman tulee esittää kuvaruudulla myös luvuista suurin ja pienin sekä niin ikään lukujen mediaani.

* 10.8 Tutki seuraavaa esimerkkiohjelmaa, joka esittää tähtitieteellistä laskelmaa 'pettämättömällä täsmällisyydellä'. Ohjelma on laadittu käyttämällä ASCII-aritmetiikkaa.

```
100 REM OHJELMAN NIMI: ASCII
110 EXTEND
120 PRINT CHR-ö-(12)
130 PRINT "AJATUSLEIKKI"
140 PRINT "= = = = = = = ="
150 PRINT
160 Valonnopeus-ö- = "299792500"
170 Vuosi-ö- = "31536000"
180 Valovuosi-ö- = MUL-ö-(Valonnopeus-ö-,Vuosi-ö-,0)
190 PRINT "VALON NOPEUS ON NOIN " Valonnopeus-ö- "M/S"
200 PRINT "JA VUOSI ON NOIN " Vuosi-ö- " SEKUNTIA"
210 PRINT
220 PRINT "ETÄISYYS MAASTA LINNUNRADAN"
230 PRINT "KESKUKSEEN ON NOIN 33000 VALOVUOTTA, TS."
240 PRINT
250 PRINT " NOIN " MUL-ö-("30000",Valovuosi-ö-,0) " METRIÄ"
260 PRINT
270 END
```

11

Graafisia mahdollisuuksia

Eri tavoilla esitetty data

Tietokoneet tulostavat useimmiten dataa numeroiden ja kirjainten muodossa, ns. aakkosnumeerisina merkkeinä. Nykyaikaisilla tietokoneilla voi dataa lisäksi havainnollistaa graafisesti, kuten esimerkiksi värillisinä kuvina kuvaruudulla. Ihminen kykenee huomattavasti paremmin käsittämään kuvia kuin aakkosnumeerisin merkein esitettyä tietoa. Siksi onkin tärkeätä, että ohjelmia laadittaessa käytetään tietokoneen graafisia mahdollisuuksia mahdollisimman tehokkaasti.

Tässä viimeisessä luvussa selvitämme niitä mahdollisuuksia, joita ei vielä käsitelty luvussa 5. Tämän luvun lopussa edellytetään, että tietokoneessa on värillinen kuvaruutu ja valmius hienografiikan käyttöön.

Erilaisia graafisia mahdollisuuksia

Yksi tapa kuvata tietokoneen graafisia ominaisuuksia on ilmoittaa miten monta paikkaa voidaan käyttää kuvaa muodostettaessa. Luvussa 5 käytimme erikoisfunktiota CUR(R,S) ilmoittamaan tulostettavan merkin sijainnin. Tässä tapauksessa sallittuja paikkoja on 960 tai 1920 käytettävästä tietokoneesta riippuen.

Nyt esitämme kaksi muuta tapaa, joilla saadaan aikaan parempia kuvia. Ensimmäinen on ns. karkea grafiikka, jossa on käytettävissä 5616 merkkipaikkaa, ja toinen hienografiikka, jossa on 57 600 paikkaa.

Pylväsdiagrammin tulostus

Seuraavalla sivulla olevassa esimerkkiohjelmassa käytetään erikoisfunktiota CUR(R,S) merkin tulostuspaikan määräämisessä. Ohjelma esittää 7 kokonaislukua pylväsdiagrammin pylväinä.

```
100 REM OHJELMA                              : PYLVÄS
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "KOKONAISLUVUT ESITETÄÄN PYLVÄSDIAGRAMMINA"
140 PRINT "= = = = = = = = = = = = = = = = = = = = = = = = = = = = ="
150 PRINT
160 PRINT "ANNA 7 KOKONAISLUKUA (PYLVÄÄN KORKEUTTA)"
170 PRINT "VÄLILLÄ 0 < KORKEUS < 20"
180 PRINT
190 REM ------- SYÖTTÖ ----------
200 FOR Indeksi = 1 TO 7
210  PRINT "ANNA PYLVÄÄN KORKEUS " Indeksi;
220  INPUT " ", Pylväskorkeus(Indeksi)
230 NEXT Indeksi
240 REM ------- PIIRRÄ PYLVÄSDIAGRAMMI ----
250 PRINT CHR$(12);
260 FOR Indeksi = 1 TO 7
270  FOR Pylväs = 1 TO Pylväskorkeus(Indeksi)
280    PRINT CUR(20—Pylväs,Indeksi*5);"
290  NEXT Pylväs
300 NEXT Indeksi
310 REM
320 REM ------- TULOSTA ASTEIKKO -------
330 PRINT CUR(0,0) " 20—"
340 PRINT CUR(10,0) " 10—"
350 PRINT CUR(20,0);
360 PRINT "   1   2   3   4   5   6   7"
370 PRINT
380 END
```

Kappaleessa PYLVÄSDIAGRAMMIN PIIRTÄMINEN piirretään pylväät kuvaruudun riviltä 20 alkaen ylöspäin. Pylväiden välinen etäisyys on 5 saraketta.

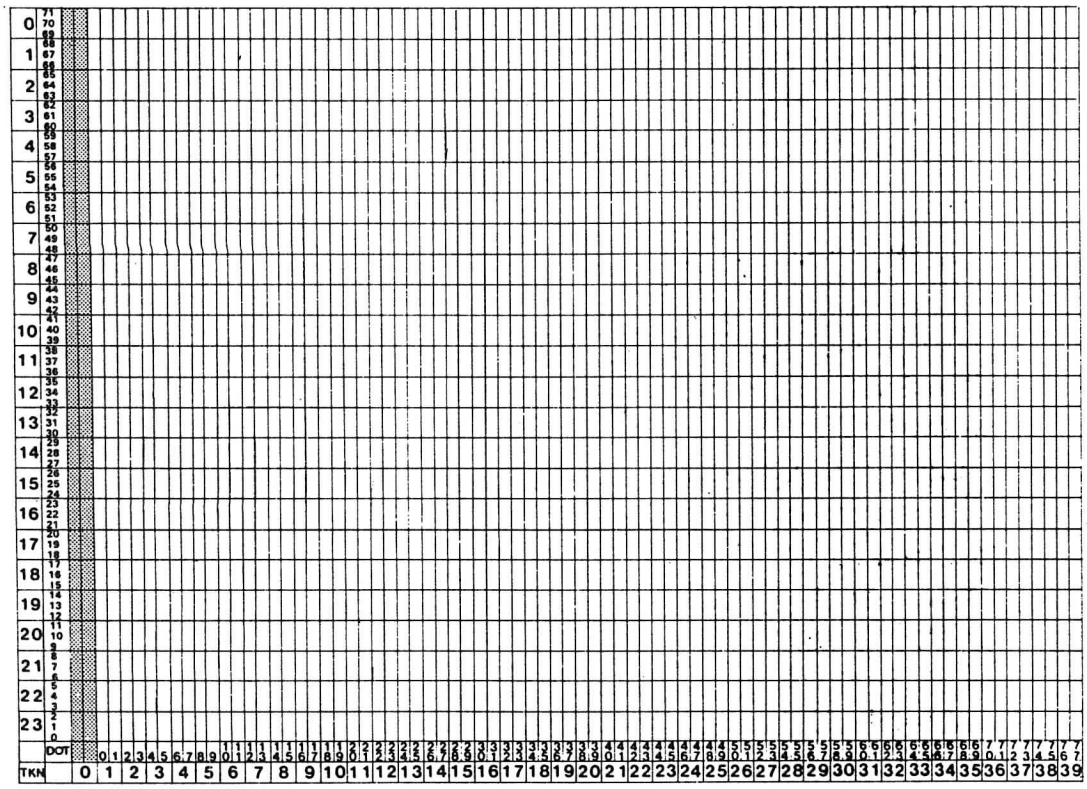
Karkean grafiikan alustus

Kuten luvussa 5 kerrottiin, täytyy erityinen ohjausmerkki kirjoittaa niille kuvaruudun riveille, joille aiotaan tulostaa graafisia merkkejä. Tällä ohjelmakappaleella sijoitetaan jokaisen kuvaruudun rivin sarakkeeseen 0 punaisen grafiikan ohjausmerkki.

```
120 REM --- ASETETAAN PUNAINEN GRAFIIKKA ---
130 PRINT CHR$(12)
140 FOR Rivi = 0 TO 23
150  PRINT CUR(RIVI,0) GRED;
160 NEXT Rivi
```

Koko kuvaruutu on nyt alustettu punaisten graafisten merkkien tulostusta varten. Graafiset merkit on lueteltu yhteenvedossa sivulla 77.

Kun kuvaruutu on asetettu grafiikkatilaan, voidaan yksi kuvaruudun 5616 pisteestä sytyttää käskyllä TXPOINT X, Y, missä X ja Y ovat sytytettävän pisteen koordinaatit. X voi saada arvon välillä 0—77 ja Y välillä 0—71.



— paikka grafiikan ohjausmerkille

Ensimmäisenä esimerkkinä käskyn TXPOINT ..., ... käytöstä piirretään kaksi akselia, joita tullaan jatkossa käyttämään funktion esityksessä. Täydennä edellinen ohjelmakappale seuraavasti.

```
100 REM OHJELMA                                  : AKSELIT
110 EXTEND
120 REM ------- ASETETAAN PUNAINEN GRAFIIKKA -------
130 PRINT CHR$(12)
140 FOR Rivi=0 TO 23
150  PRINT CUR(Rivi,0) GRED;
160 NEXT Rivi
170 REM ------- VAAKASUORA AKSELI -------
180 FOR X=0 TO 77
190  TXPOINT X,35
200 NEXT X
210 REM ------- PYSTYSUORA AKSELI -------
220 FOR Y=10 TO 60
230  TXPOINT 0,Y
240 NEXT Y
250 REM ------- LOPETUS ----------
260 PRINT CUR(0,0); "AKSELIT"
270 END
```

Funktion esitys karkealla grafiikalla

Edellisessä luvussa luotiin ohjelmalla FUNKTIED peräkkäistiedosto nimeltä LUKUPARI.DAT. Tiedosto sisältää lukuparit (X, Y), missä Y = SIN(X)*EXP(−.3*X). Ohjelmalla KARGRAF1 voidaan nämä lukuparit esittää karkeata grafiikkaa käyttäen.

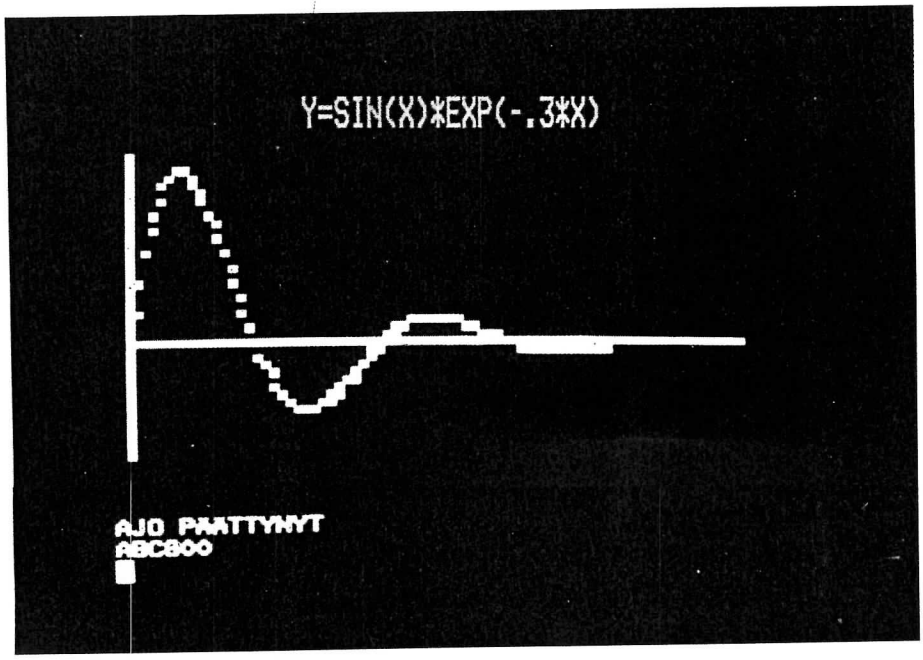
Ohjelmakappale, joka alustaa kuvaruudun karkeata grafiikkaa varten, on määritelty funktiona FNKelgraf$. Myös koordinaattiakselit piirtävä kappale on määritelty funktiona, FNAkselit$.

Koska todellisuudessa X saa arvot välillä 0—4X, on nämä arvot kerrottu skaalauskertoimella 5 kuvion venyttämiseksi vaakasuorassa suunnassa sopivammaksi. Y-arvoille käytetään skaalauskerrointa 35. Koska origo sijaitsee pisteessä (0, 35), lisätään Y-arvoihin vielä 35.

```
100 REM OHJELMA                                  : KARGRAF1
110 EXTEND
120 PRINT FNKeltgraf$; : REM KUVARUUTU GRAFIIKKATILAAN
130 PRINT FNAkselit$; : REM PIIRRETÄÄN AKSELIT
140 REM
150 REM ------- LUE TIEDOSTOSTA JA PIIRRÄ
160 OPEN "LUKUPARI.DAT" AS FILE 1
170 ON ERROR GOTO 270
180 INPUT #1, Xarvo
```

```
190 INPUT #1, Yarvo
200 TXPOINT 5*Xarvo;35+35*Yarvo
210 GOTO 180
220 CLOSE 1
230 REM ------- TULOSTUS, LOPETUS ----
240 PRINT CUR(1,10) RED DBLE "Y=SIN(X)*EXP(—.3*X)"
250 PRINT CUR(20,0) "AJO PÄÄTTYNYT"
260 END
270 REM ------- VIRHEEN KÄSITTELIJÄ
280 RESUME 220
290 REM ------- KELTGRAF.DEF -------
300 DEF FNKeltgraf$ LOCAL Rivi
310  PRINT CHR$(12)
320  Rivi=0
330  WHILE Rivi<23
340     PRINT CUR(Rivi,0) GYEL;
350     Rivi=Rivi+1
360  WEND
370  RETURN Eimitään$
380 FNEND
390 REM ------- AKSELIT.DEF -------
400 DEF FNAkselit$ LOCAL X,Y
410  X=0 : Y=20
420  WHILE X<78
430     TXPOINT X,35
440     X=X+1
450  WEND
460  WHILE Y<61
470     TXPOINT 0,Y
480     Y=Y+1
490  WEND
500  RETURN Eimitään$
510 FNEND
```

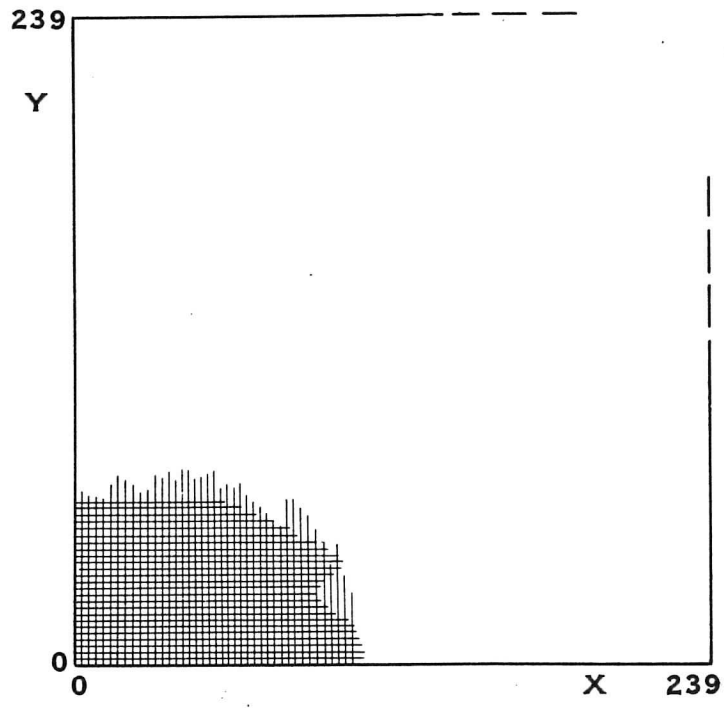
Ajon jälkeen kuvaruutu näyttää tältä.



Huomaa, että silmukat on kummassakin funktion määrittelyssä toteutettu käskyillä WHILE ... ja WEND. Tämä johtuu siitä, että ilmauksen FOR ... TO ... silmukkamuuttujaa ei saa määritellä paikalliseksi muuttujaksi. Katso TXPOINT-käskyn käyttöä tarkemmin yhteenvedosta.

Hienografiikka

Kun tietokone varustetaan hienografiikkaa varten tarvittavalla laitteistolla, parannetaan koneen graafisia ominaisuuksia huomattavasti. Hienografiikkaa käytettäessä mahdollisten kuvapisteiden lukumäärä nousee 5616:sta 57 600 pisteeseen. Alla olevasta kuviosta ilmenee, miten pisteiden sijainti kuvaruudulla annetaan.



Hienografiikan ohjauskäskyt ovat:

FGCTL ...
FGPOINT X, Y, ...
FGLINE X, Y, ...
FGFILL X, Y, ...
FGPAINT X, Y, ...

Kuvion koordinaatit annetaan X:n ja Y:n paikalla. Näiden sallitut arvot näkyvät yllä olevasta kuviosta.

Hienografiikan erillinen muisti

Hienografiikan käskyillä muodostetut kuvat talletetaan erilliseen kuvamuistiin. Tietokoneen tavallisen kuvamuistin sisältö eli graafiset merkit ja karkea grafiikka voidaan esittää riippumatta hienografiikasta.

Hienografiikkaa voidaan esittää neljällä eri värillä yhtaikaa ja kuvat voivat sisältää siis korkeintaan neljää väriä. Koska tietokoneessa on kuitenkin valittavissa 8 eri väriä, joissa on mukana musta ja valkoinen, täytyy jokaista kuvaa varten valita neljän värin valikoima. Tämä tapahtuu värivalintakäskyllä FGCTL ... (engl. Fine Grafics ConTroL). Värikombinaatioita on kaikkiaan 128 ja esimerkiksi käsky FGCTL 3 antaa "värit" musta (0), punainen (1), vihreä (2) ja sininen (3). Suluissa olevat luvut ilmoittavat vastaavan värinumeron yhdistelmässä. Käytä seuraavaa käskyä komentona ja katso mitä tapahtuu.

```
FGCTL 3
```

Kuten huomaat, täytyy kuvamuisti tyhjentää ennen kuin uusi kuva voidaan tallettaa. Seuraava rutiini, joka tyhjentää kuvamuistin, sisältää värivalinnan FGCTL 0, joka tarkoittaa että kaikki värivalinnan neljä väriä ovat mustia. Ohjelma kirjoittaa myös värinumeron 0 joka paikkaan kuvamuistissa.

Komennon CLEAR antaminen lopettaa hienografiikan kuvamuistin esittämisen kuvaruudulla. Anna siksi komento CLEAR ennen seuraavan ohjelman syöttöä.

```
10 REM ------- TYHJENNÄ -------
20 PRINT CHR$(12)
30 FGCTL 0
40 FGPOINT 0,0,0
50 FGFILL 239,239
```

Ohjelman ajon jälkeen voit todeta kuvamuistin tyhjäksi käyttämällä uudestaan värivalintakäskyä FGCTL 3. Rutiinissa TYHJENNÄ esiintyvien käskyjen FGPOINT ja FGFILL merkitys selitetään seuraavilla sivuilla.

Hienografiikan käyttö

Koska kuvamuisti täytyy yleensä tyhjentää useammin kuin kerran ohjelmassa, määritellään kappale TYHJENNÄ funktiona FNTyhjennä$. Funktio poistaa rivillä 390 myös tekstin ja karkean grafiikan tavallisesta kuvamuistista.

```
370 REM ------- TYHJENNÄ.DEF -------
380 DEF FNTyhjennä$
390  PRINT CHR$(12)
400  FGCTL 0
410  FGPOINT 0,0,0
420  FGFILL 239,239
430  RETURN Eimitään$
440 FNEND
```

Ohjelma aloitetaan nyt molempien kuvamuistien tyhjennyksellä rivillä 120. Rivillä 140 valitaan väriyhdistelmä 3, jolloin käytettävissä ovat värit musta (0), punainen (1), vihreä (2) ja sininen (3).

```
100 REM OHJELMA                          : HIEGRAF1
110 EXTEND
120 PRINT FNTyhjennä$
130 PRINT "VÄRIVALINTAKOMENTO 3"
140 FGCTL 3
150 PRINT
```

Käskyllä FGPOINT X, Y, V voidaan sytyttää piste, jonka koordinaatit ovat X ja Y, ja asettaa siihen värinumero V vastaava väri. Täydennä ohjelma ja aja se.

```
160 PRINT "ORIGO (0,0), PUNAINEN PISTE (1)"
170 FGPOINT 0,0,1
```

Käskyä FGLINE X, Y, V käytetään piirtämään värillä V suora edellisestä pisteestä pisteeseen, jonka koordinaatit ovat X ja Y.

```
180 PRINT
190 PRINT "(0,0):STA (239,0):AAN PUNAINEN VIIVA (1)"
200 FGLINE 239,0,1
210 PRINT
220 PRINT "(239,0):STA (239,239):ÄÄN VIHREÄ VIIVA (2)"
230 FGLINE 239,239,2
240 PRINT
250 PRINT "(239,239):STÄ (0,0):AAN SININEN VIIVA (3)"
260 FGLINE 0,0,3
```

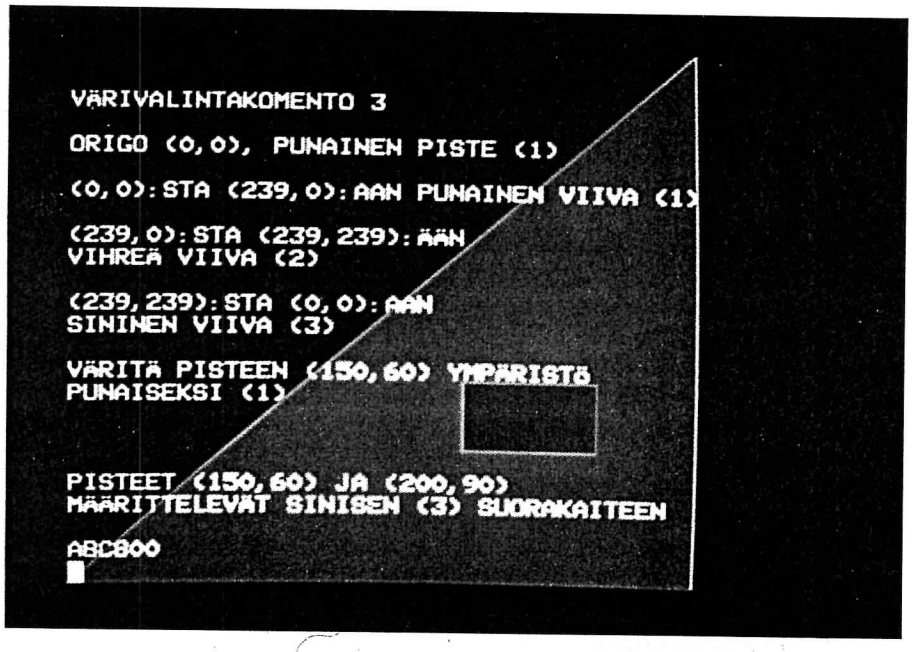
Käskyllä FGPAINT X, Y, V voidaan suljettu alue värittää värillä V. X ja Y ovat valinnaiset koordinaatit alueen sisällä.

```
270 PRINT
280 PRINT "VÄRITÄ PISTE (150,60)"
290 PRINT "PUNAISEKSI (1)"
300 FGPAINT 150,60,1
```

Haluttaessa esittää suorakaiteen muotoinen pinta värillä V siten, että yksi kulma on edellisessä ilmoitetussa pisteessä ja vastakkainen kulma pisteessä (X, Y), käytetään käskyä FGFILL X, Y, V.

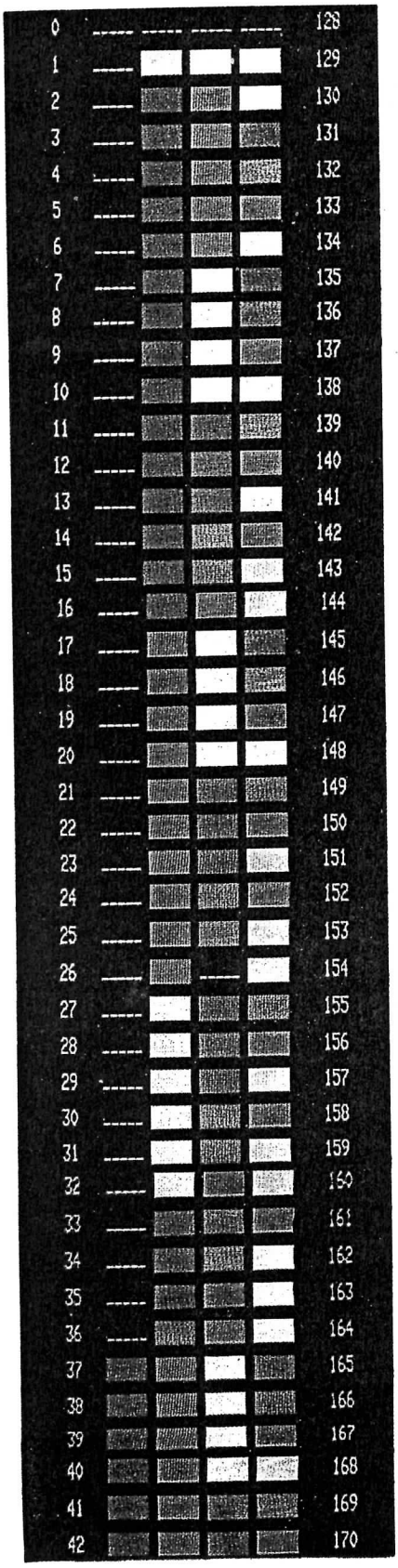
```
310 PRINT
320 PRINT CUR(19,0) "PISTEET (150,60) JA (200,90)"
330 PRINT "MÄÄRITTELEVÄT SINISEN (3) SUORAKAITEEN"
340 FGFILL 200,90,3
350 PRINT
360 END
```

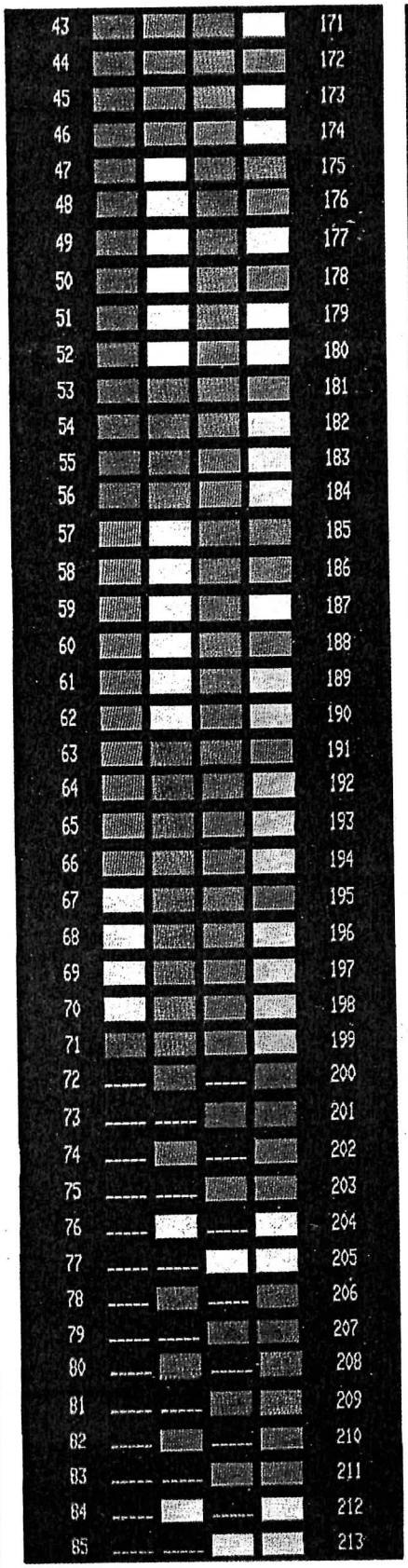
Kun koko ohjelma on ajettu, näyttää kuvaruutu seuraavalta:

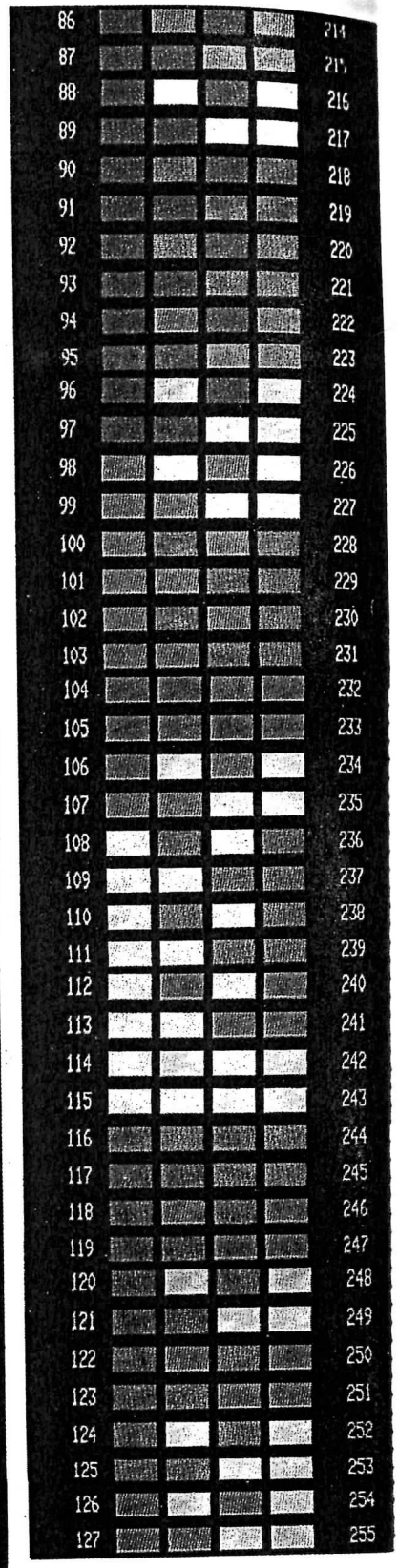


Mahdolliset väriyhdistelmät

Värinumerolla on eri merkityksiä tehdystä värivalinnasta riippuen. Sinun kannattaa kokeilla edellistä ohjelmaa eri värivalinnoilla, esim. 33 tai 37, muuttamalla ohjelmariviä 140. Värivalintoja 0—127 käytettäessä voidaan kummankin kuvamuistin sisällöt esittää yhtaikaa. Jos sitävastoin halutaan esittää vain hienografiikkamuistin sisältö, käytetään värivalintoja 128—255. Seuraavassa taulukossa on näkyvissä eri värivalinnoissa käytettävät värit.


0 128
1 129
2 130
3 131
4 132
5 133
6 134
7 135
8 136
9 137
10 138
11 139
12 140
13 141
14 142
15 143
16 144
17 145
18 146
19 147
20 148
21 149
22 150
23 151
24 152
25 153
26 154
27 155
28 156
29 157
30 158
31 159
32 160
33 161
34 162
35 163
36 164
37 165
38 166
39 167
40 168
41 169
42 170


43 171
44 172
45 173
46 174
47 175
48 176
49 177
50 178
51 179
52 180
53 181
54 182
55 183
56 184
57 185
58 186
59 187
60 188
61 189
62 190
63 191
64 192
65 193
66 194
67 195
68 196
69 197
70 198
71 199
72 200
73 201
74 202
75 203
76 204
77 205
78 206
79 207
80 208
81 209
82 210
83 211
84 212
85 213


86 214
87 215
88 216
89 217
90 218
91 219
92 220
93 221
94 222
95 223
96 224
97 225
98 226
99 227
100 228
101 229
102 230
103 231
104 232
105 233
106 234
107 235
108 236
109 237
110 238
111 239
112 240
113 241
114 242
115 243
116 244
117 245
118 246
119 247
120 248
121 249
122 250
123 251
124 252
125 253
126 254
127 255

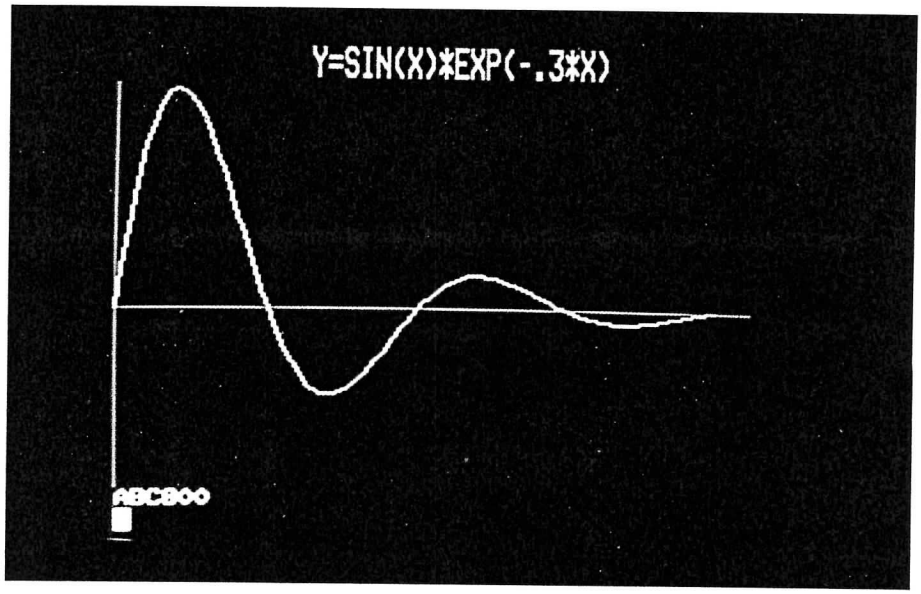
Kuviot hienografiikalla

Peräkkäistiedostoon LUKUPARI.DAT talletettuja lukupareja (X, Y) havainnollistetaan nyt hienografiikalla. Tällä ohjelmalla on sama rakenne kuin ohjelmalla KARGRAF1.

```
100 REM OHJELMA                          : HIEGRAF2
110 EXTEND
120 REM DATA LUETAAN TIEDOSTOSTA LUKUPARI.DAT
130 REM JA TULOSTETAAN KUVARUUDULLE
140 PRINT FNTyhjennä$;
150 FGCTL 7 : REM VÄRIKOMBINAATIO 7
160 PRINT FNAkselit$;
170 REM
180 REM ------- LUKU JA TULOSTUS ----------
190 OPEN "LUKUPARI.DAT" AS FILE 1
200 Xskaalauskerroin=18 : Yskaalauskerroin=150
210 Xpiste=0 : Ypiste=100
220 ON ERROR GOTO 330
230 INPUT #1, X
240 INPUT #1, Y
250 FGPOINT Xskaalauskerroin*X,Ypiste+Yskaalauskerroin*Y,2
260 GOTO 230
270 CLOSE 1
280 REM -- KIRJOITA OTSIKKO JA LOPETA
290 PRINT CUR(3,10) GRN DBLE "Y=SIN(X)*EXP(—.3*X)"
300 PRINT CUR(22,0)
310 END
320 REM ------- VIRHEEN KÄSITTELIJÄ
330 RESUME 270
340 REM ------- TYHJENNÄ.DEF -------
350 DEF FNTyhjennä$
360  PRINT CHR$(12)
370  FGCTL 0
380  FGPOINT 0,0,0
390  FGFILL 239,239
400  RETURN Eimitään$
410 FNEND
420 REM ------- AKSELIT.DEF -------
430 DEF FNAkselit$
440  FGPOINT 0,100,1
450  FGLINE 239,100
460  FGPOINT 0,20
470  FGLINE 0,200
480  FGPOINT 0,100 : REM ORIGO
490  RETURN Eimitään$
500 FNEND
```

Luvun 10 ohjelmaa MAXARVO voidaan käyttää laskemaan skaalauskerroin automaattisesti. Tietysti myös funktio Akselit$ voidaan tehdä yleisemmäksi siten, että koordinaattiakselien sijainnit

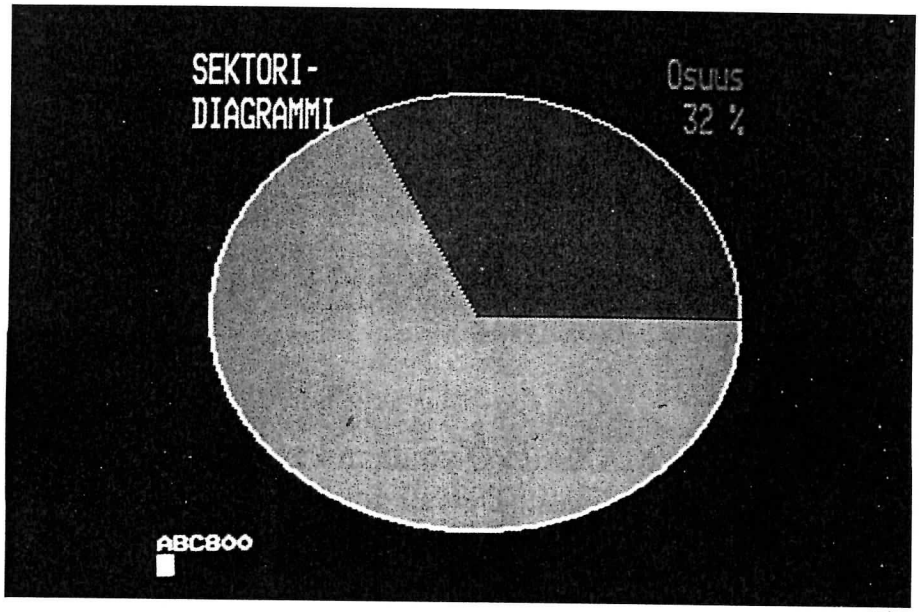
ilmoitetaan muuttujilla. Kuten riveiltä 450—480 näkyy, voidaan värinumero jättää ilmoittamatta. Tällöin viimeksi ilmoitettu värinumero on voimassa, tässä tapauksessa rivillä 440 annettu väri 1.



Kun riville 250 muutetaan käsky FGLINE, saadaan kuvan mukainen jatkuva käyrä.

Sektoridiagrammi

Viimeisenä esimerkkinä hienografiikan käytöstä annetaan ohjelma, joka piirtää kuvan mukaisen sektorin sektoridiagrammiin.



Ohjelma on tässä:

```
100 REM OHJELMA                          : HIEGRAF3
110 EXTEND
120 SINGLE
130 PRINT CHR$(12)
140 INPUT "ANNA OSUUS PROSENTEISSA",Osuus
150 Kulma=2*PI*Osuus/100
160 PRINT FNTyhjennä$
170 FGCTL 9 : REM VÄRIYHDISTELMÄ 9
180 REM
190 REM ------- PIIRRÄ YMPYRÄ -------
200 FGPOINT 220,120,0 : REM LÄHTÖPISTE
210 FOR X=0 TO 6.3 STEP PI/100
220  FGLINE 120+100*COS(X),120+100*SIN(X),2
230 NEXT X
240 REM ------- VÄRITÄ YMPYRÄ -----------
250 FGPAINT 120,120,3 : REM KESKIPISTE
260 REM
270 REM ------- PIIRRÄ SEKTORI -------
280 FGPOINT 220,120,0 : REM LÄHTÖPISTE
290 FGLINE120,120 : REM KESKIPISTE
300 FGLINE 120+100*COS(Kulma),120+100*SIN(kulma)
310 FGPAINT 120+50*COS(Kulma/2),120+50*SIN(Kulma/2),1
320 REM
330 REM --- KIRJOITA OTSIKKO JA LOPETA -
340 PRINT CUR(1,0) DBLE YEL "SEKTORI-";
350 PRINT CUR(3,0) DBLE YEL "DIAGRAMMI";
360 PRINT CUR(1,30) DBLE RED "Osuus"
370 PRINT CUR(3,30) DBLE RED Osuus "%"
380 PRINT CUR(22,0)
390 END
400 REM ------- TYHJENNÄ.DEF -------
410 DEF FNTyhjennä$
420  PRINT CHR$(12)
430  FGCTL 0
440  FGPOINT 0,0,0
450  FGFILL 239,239
460  RETURN Eimitään$
470 FNEND
```

Tässä luvussa olemme voineet viitata vain muutamiin graafisen tuostuksen sovellusalueisiin. Lisää esimerkkejä on HARJOITUSTEHTÄVISSÄ.

Yhteenveto

Väri ja grafiikka ABC 800:lle

TXPOINT X, Y, S
Sytyttää (S = 1 tai S jätetty pois) tai sammuttaa (S = 0) graafisen pisteen, jonka koordinaatit ovat X ja Y. Kohdan on oltava graafisessa tilassa.
0 < = X < = 77 0 < = Y < = 71
80 TXPOINT 77,71 ! YLHÄÄLLÄ OIKEALLA

TXPOINT (X, Y)
Antaa arvon tosi (−1), jos kuvaruudun piste (X, Y) on sytytetty ja muuten arvon EPÄTOSI (0).
10 IF TXPOINT (X, Y) THEN TXPOINT X, Y,
0 ELSE TXPOINT X, Y

Hienografiikka

FGCTL K
Valitsee värikombinaation K, missä
0 < = K < = 255.

FGFILL X, Y, V
Värittää suorakaiteen edellisestä pisteestä pisteeseen (X, Y) värillä V.

FGLINE X, Y, V
Piirtää suoran edellisestä pisteestä pisteeseen (X, Y) värillä V.

FGPOINT X, Y, V
Värittää pisteen (X, Y) värillä V.

FGPOINT (X, Y)
Antaa pisteen (X, Y) värinumeron.

Harjoitustehtäviä

* 11.1 Muuta ohjelmaa PYLVÄÄT siten, että pylväsdiagrammissa voidaan esittää 30 lukua välillä 0—100.

* 11.2 Karkean grafiikan avulla voidaan ohjelmaa PYLVÄÄT muuttaa siten, että diagrammissa pylväät esitetään sinisinä keltaista taustaa vasten ja luvut esitetään punaisella.

* 11.3 Muuta ohjelmaa PYLVÄÄT siten, että kuvat esitetään hienografiikalla ja pylväiden korkeus on välillä 0—200.

* 11.4 Muuta seuraavaa ohjelmaa siten, että spiraali esitetään keltaisena punaisella pohjalla.

```
100 REM OHJELMA                          : L11H4
110 EXTEND
120 PRINT FNGraftila$
130 FOR X=0 TO 60 STEP PI/100
140  TXPOINT 35+X*COS(X)/2,35+X*SIN(X)/2
150 NEXT X
160 PRINT CUR(0,0);
170 END
180 REM ------- GRAFTILA.DEF -------
190 DEF FNGraftila$ LOCAL Rivi
200  PRINT CHR$(12)
210  Rivi=0
220  WHILE Rivi<23
230     PRINT CUR(Rivi,0) YEL NWBG GYEL;
240     Rivi=Rivi+1
250  WEND
260  RETURN Eimitään$
270 FNEND
```

* 11.5 Kirjoita ohjelma, joka piirtää funktioiden Y1 = SIN(X) ja Y2 = COS(X) kuvaajat välillä $0 <= X <= 2\pi$ käyttäen hienografiikkaa.

*** 11.6 Tämän ohjelman hienografiikalla voidaan kuvata prosessin toimintaa. Aja ohjelma ja tarkastele sen toimintaa.**

```
100 REM OHJELMA                                    : L11H6
110 EXTEND
120 PRINT FNTyhjennä$
130 FGCTL 3
140 REM ------- PIIRRÄ TANKKI 1 -------
150 FGPOINT 0,0,1
160 FGLINE 0,110
170 FGLINE 60,110
180 FGLINE 60,0
190 FGLINE 0,0
200 REM ------- PIIRRÄ TANKKI 2 -------
210 FGPOINT 200,0
220 FGLINE 230,0
230 FGLINE 230,220
240 FGLINE 200,220
250 FGLINE 200,0
260 REM ------- PIIRRÄ PUTKI -------
270 FGFILL 60,5,1
280 REM ------- TÄYTÄ TANKKI 1 -------
290 FGPOINT 1,1,3
300 FGFILL 59,109,3
310 REM ------- PUMPPAA ----------
320 PRINT CUR(0,0) DBLE GRN "Pumppa=Näppä"
330 FOR Y=1 TO 103
340  GET Pumppaa$
350  PRINT CUR(22,17) YEL "PUMPPAA"
360  FGPOINT 1,109,0
370  FGFILL 59,109-Y,0
380  FGPOINT 201,1,0
390  FGFILL 229,2*Y,3
400  REM ------- TESTAA ----------
410  IF FGPOINT(201,200)=3 THEN PRINT CUR(0,0) RED DBLE "Yläraja ---"
420  IF FGPOINT(201,200)=0 THEN PRINT CUR(0,0) RED DBLE "         ---"
430  PRINT CUR(22,17) SPACE$(10)
440 NEXT Y
450 REM ------- LOPETUS ----------
460 END
470 REM ------- TYHJENNÄ.DEF -------
480 DEF FNTyhjennä$
490  PRINT CHR$(12)
500  FGCTL 0
510  FGPOINT 0,0,0
520  FGFILL 239,239
530  RETURN Eimitään$
540 FNEND
```

11.7 Liitä ohjelmassa HIEGRAF2 määritellyt funktiot FNTyhjennä$ ja FNAkselit$ seuraavaan ohjelmaan. Aja sitten ohjelma ja tutki tuloksia.

```
100 REM OHJELMA                          : L11H7
110 EXTEND
120 SINGLE
130 PRINT CHR$(12)
140 PRINT "'NELIÖAALLON' FOURIERSARJAN TERMIEN"
150 PRINT "SUMMAN ESITYS"
170 PRINT
180 INPUT "MITEN MONTA TERMIÄ SUMMATAAN ?"Määrä
190 PRINT FNTyhjennä$
200 FGCTL 7 : REM VÄRIYHDISTELMÄ 7
210 PRINT FNAkselit$;
220 PRINT CUR(5,20) DBLE RED "TERMIEN MÄÄRÄ " Määrä
230 Amplitudi=1
240 FOR X=0 TO 6.3 STEP PI/100
250  Summa=0
260  FOR K=1 TO 2*Määrä STEP 2
270   LET Summa=Summa+Amplitudi/K*SIN(K*X)
280  NEXT C
290  FGLINE 30*X,100+100*Summa,2
300 NEXT X
310 PRINT CUR(21,0) GRN DBLE "AJO PÄÄTTYNYT"
320 END
```

11.8 Edistyksellinen taidegallerian omistaja asetti seuraavalla ohjelmalla varustetun tietokoneen seinälle. Aja ohjelma ja arvostele tietokoneen taiteellinen lahjakkuus.

```
100 REM OHJELMA                                  : L11H8
110 EXTEND
120 SINGLE
130 PRINT FNTyhjennä$
140 RANDOMIZE
141 FOR Kuva=1 TO 20
150  FGCTL 127*RND
160  FGPOINT RND*239,RND*239,RND*4
170  FGLINE RND*239,RND*239,RND*4
180  FGLINE RND*239,RND*239,RND*4
190  FGLINE RND*239,RND*239,RND*4
200  FGPAINT RND*239,RND*239,RND*4
210  FOR Kierros=1 TO 500+1000*RND : NEXT Kierros
220 NEXT Kuva
222 GOTO 130
230 REM ------- TYHJENNÄ.DEF -------
240 DEF FNTyhjennä$
250  PRINT CHR$(12)
260  FGCTL 0
270  FGPOINT 0,0,0
280  FGFILL 239,239
290  RETURN Eimitään$
300 FNEND
```

12

Harjoitus-tehtävien vastaukset

2.1 —

2.2 Tulostuksen alkuun ja loppuun tulee tyhjä rivi.

2.3
```
 5 PRINT
10 PRINT "TÄMÄ"
20 PRINT "ON TOINEN"
30 PRINT "YRITYS"
40 PRINT
```

2.4
```
10 PRINT "*************************"
20 PRINT "*        DIPLOMI        *"
30 PRINT "*                       *"
40 PRINT "*   MAIJA SUOMALAINEN   *"
50 PRINT "*                       *"
60 PRINT "*************************"
```

2.5 Pilkut aikaansaavat tulostuksen sarakkeissa.

2.6
```
 5 PRINT
10 PRINT "TUOTE", "LUKUMÄÄRÄ", "HINTA"
15 PRINT
20 PRINT "KAULAKETJU", "3", "225,—"
30 PRINT "PIIKKARIT", "7", "340,—"
40 PRINT "ONKIMATO", "12", "2,—"
50 PRINT
```

2.7 Puolipiste PRINT-lauseissa antaa 'tiheän' tulostuksen. Kun PRINT-lause loppuu puolipisteeseen, ei PRINT-lausetta ajettaessa tapahdu rivinsiirtoa.

2.8 Tulostus sijoittuu toiseen sarakkeeseen.

2.9 —

3.1
```
10 LET A$ = "OSAAN"
20 LET B$ = "MINÄ"
30 LET C$ = "OHJELMOIDA"
40 PRINT B$,A$,C$
```

3.2
```
10 INPUT "KIRJOITA KAUPUNGIN NIMI: "A$
20 WHILE N<60
30  N=N+1
40  PRINT A$,
50 WEND
```

3.3
```
10 INPUT "KIRJOITA KAUPUNGIN NIMI: "A$
20 WHILE N<20
30  N=N+1
40  PRINT A$ "ON KAUNIS KAUPUNKI"
50 WEND
```

3.4

```
10 PRINT "MINKÄ NIMINEN ON NORJAN"
20 INPUT "PÄÄKAUPUNKI? "S$
30 PRINT
40 IF S$="OSLO" THEN PRINT "OIKEA KAUPUNKI"
50 IF S$<>"OSLO" THEN PRINT "VÄÄRÄ MAA"
60 PRINT
```

3.5

```
10 INPUT "KIRJOITA NIMI/(LOPPU): "S$
20 PRINT
30 PRINT, S$
40 PRINT
50 IF S$<>"LOPPU" THEN GOTO 10
```

3.6

```
100 PRINT
110 PRINT "SYÖTÄ VANHOJA AMMATTINIMIKKEITÄ"
120 WHILE A$(N)<>"LOPPU"
130  LET N=N+1
140  INPUT "NIMIKE (/LOPPU): "A$(N)
150 WEND
160 LET N=1
170 PRINT
180 PRINT
190 PRINT "YHTIÖN UUSIA TITTELEITÄ"
200 PRINT
210 WHILE A$(N)<>"LOPPU"
220  PRINT "BYRO" A$(N)
230  LET N=N+1
240 WEND
```

3.7 —

4.1

```
200 PRINT CHR$(12)
220 PRINT "JAKOLASKU"
240 PRINT "= = = = = = = = = ="
260 PRINT
280 INPUT "ANNA ENSIMMÄINEN LUKU "A
300 INPUT "ANNA TOINEN LUKU      "B
320 PRINT
340 PRINT "LUKUJEN OSAMÄÄRÄ ON "A/B
360 PRINT CUR(18,0);
380 INPUT "HALUATKO JATKAA (KYLLÄ/EI) "S$
400 IF S$<>"EI" THEN GOTO 200
420 PRINT "AJO SUORITETTU"
```

4.2

```
100 PRINT CHR$(12)
110 INPUT "ANNA ALKUPERÄINEN HINTA : "A
120 INPUT "ANNA ALENNUSPROSENTTI : "P
130 PRINT
140 PRINT "ALENNETTU HINTA: ";
150 PRINT USING "# # # # # #.# #" A—A*P/100
160 PRINT
170 INPUT "UUSI AJO (KYLLÄ/EI): "V$
180 IF V$<>"EI" THEN GOTO 100
190 PRINT
```

4.3

```
10 PRINT CHR$(12)
20 INPUT "ANNA SALDO, JOSTA LASKENTA ALOITETAAN (MK) "S
30 PRINT
40 PRINT "UUSI SALDO", "TAPAHTUMA"
50 PRINT
60 PRINT S,
70 INPUT " "T
80 S=S+T
90 GOTO 60
```

4.4

```
10 PRINT CHR$(12)
12 PRINT "LASKELMA"
13 PRINT "= = = = = = = = ="
20 INPUT "PALKANKOROTUS/KK     (MK) : "L
30 INPUT "MARGINAALIVERO        (%) : "M
40 INPUT "TYÖTUNTEJA/KK             : "T
45 PRINT
50 PRINT "PALKANKOROTUS/TUNTI       :";
60 PRINT USING "# #.# #" (L—M*L/100)/T;
70 PRINT "MARKKAA"
80 PRINT
```

4.5

```
10 PRINT CHR$(12)
20 INPUT "ANNA PÄÄOMA           (MK) : "K
30 INPUT "ANNA KORKOPROSENTTI       : "P
40 PRINT "PÄÄOMA", "KORKO%", "VUOSIA"
50 WHILE V$<>"EI"
60  K=K+K*P/100
65  A=A+1
70  PRINT CUR(5,0) K,P,A
80  PRINT
85  PRINT "HALUAN LASKELMAN"
90  INPUT "VIELÄ YHDELLE VUODELLE (KYLLÄ/EI)? "V$
100 WEND
```

4.6

```
10  PRINT CHR$(12)
20  INPUT "ANNA ALENTAMATON HINTA (MK) : "H
30  A=H
40  IF H>=50 THEN A=.9*H
50  IF H>=200 THEN A=.8*H
60  IF H>=500 THEN A=.7*H
70  IF H>=1000 THEN A=.6*H
80  PRINT : PRINT
90  PRINT "ALENNETTU HINTA (MK):";
100 PRINT USING # # # #.# #" A
```

4.7

```
100 PRINT CHR$(12)
110 PRINT "VEIKKAUSRIVI"
120 PRINT "= = = = = = = = = = = ="
130 PRINT
140 RANDOMIZE
150 FOR N=1 TO 13
160  S=INT (1+RND*5)
170  IF S<4 TEN PRINT " 1"
180  IF S=4 THEN PRINT "  X"
190  IF S=5 THEN PRINT "    2"
200 NEXT N
210 PRINT
```

4.8

```
100 PRINT CHR$(12)
110 RANDOMIZE
120 FOR N=1 TO 7
130   S(N)=INT (1+RND*37)
140   FOR K=1 TO N-1
150     IF S(N)=S(K) THEN N=N-1
160   NEXT K
170 NEXT N
180 PRINT "                    LOTTOEHDOTUS"
190 PRINT "                    ============="
200 PRINT
210 FOR N=1 TO 7
220   PRINT ,S(N)
230   PRINT
240 NEXT N
250 PRINT
```

4.9

```
100 PRINT CHR$(12)
110 PRINT "Aseta tietokoneen kello oikeaan aikaan"
120 PRINT "======================================"
130 PRINT
140 INPUT "VUOSI             (esim. 83) "V
150 INPUT "KUUKAUSI          (esim. 12) "K
160 INPUT "PÄIVÄ             (esim. 22) "P
170 INPUT "TUNTI             (esim. 08) "T
180 INPUT "MINUUTTI          (esim. 26) "M
190 INPUT "SEKUNTI           (esim. 00) "S
200 POKE -17,V,K,P,T,M,S
210 PRINT CHR$(12)
220 PRINT CUR(0,20) TIME$
230 GOTO 220
```

4.10

```
100 PRINT CHR$(12)
110 PRINT "A R V A A  L U K U"
120 PRINT "=========="
130 PRINT
140 RANDOMIZE
150 LET S=INT (RND*10)
160 PRINT
170 INPUT "ANNA ARVAUKSESI "A
180 IF S=A THEN GOTO 210
190 PRINT "VÄÄRÄ VASTAUS — YRITÄ UUDELLEEN ! "
200 GOTO 160
210 PRINT "OIKEA VASTAUS ! "
```

5.1 a

```
10 PRINT RED "VÄRILLINEN TULOSTUS"
```

b

```
10 PRINT DBLE RED "VÄRILLINEN TULOSTUS"
```

c

```
10 PRINT FLSH DBLE RED "VÄRILLINEN TULOSTUS"
```

d

```
10 PRINT YEL NWBG FLSH DBLE RED "VÄRILLINEN TULOSTUS"
```

5.2
```
100 PRINT CHR-ö-(12)
110 PRINT DBLE YEL "N O P P A P E L I"
120 PRINT "= = = = = = = = = = = = = = = = = = = = = = = ="
130 PRINT CUR(6,1) YEL " NOPAT NÄYTTÄVÄT:"
140 PRINT
150 RANDOMIZE
160 FOR N=1 TO 5
164  FOR K=1 TO 1000
166  NEXT K
170  PRINT DBLE RED INT(1+RND*6);
180 NEXT N
190 PRINT CUR(18,0);
200 INPUT " HALUATKO JATKAA ? (KYLLÄ/EI) "V$
210 IF V-ö-<>"EI" THEN GOTO 100
220 PRINT "AJO PÄÄTTYNYT"
```

5.3
```
130 A=A+RND*.3
```

5.4
```
100 PRINT CHR-ö-(12)
110 FOR K=0 TO 23
120  PRINT CUR(K,O) BLU NWBG;
130 NEXT K
140 PRINT CUR(3,2) DBLE FLSH YEL "PÄIVÄN ERIKOISTARJOUS:"
150 PRINT CUR(5,2) DBLE FLSH YEL "= = = = = = = = = = = = ="
160 PRINT CUR(10,12) DBLE YEL "MAKKARA 30,—/KG"
170 PRINT CUR(13,12) DBLE YEL "JUUSTO   25,—/KG"
180 PRINT CUR(16,12) DBLE YEL "SILLI    39,—/KG"
```

5.4
```
100 PRINT CHR-ö-(12)
110 FOR K=0 TO 23
120  PRINT CUR(K,0) BLU NWBG;
130 NEXT K
140 PRINT CUR(3,2) DBLE FLSH YEL "PÄIVÄN ERIKOISTARJOUS:"
150 PRINT CUR(5,2) DBLE FLSH YEL "= = = = = = = = = = = = ="
160 PRINT CUR(10,12) DBLE YEL "MAKKARA  30,—/KG"
170 PRINT CUR(13,12) DBLE YEL "JUUSTO   25,—/KG"
180 PRINT CUR(16,12) DBLE YEL "SILLI    39,—/KG"
```

tehtävässä: rivit 110, 120 ja 130 puuttuu

5.5
```
10 PRINT CHR$(12)
20 INPUT "ANNA ALKUPERÄINEN SALDO (MK):"S
30 PRINT CHR$(12)
40 PRINT GRN "UUSI SALDO", "MUUTOS"
50 PRINT
60 PRINT CYA S, YEL;
70 INPUT " "M
80 S=S+M
90 GOTO 60
```

5.6
```
100 PRINT CHR$(12)
110 FOR N=1 TO 10
120  PRINT CHR-ö-(12)
130  PRINT CUR(4,1) DBLE MAG "S Ä V E L M Ä  N R O " N
140  PRINT CUR(12,1) DBLE YEL ". . . . . "
150  PRINT CUR(14,1) DBLE YEL "0 1 2 3 4 5 "
160  PRINT CUR(18,1);
170  RANDOMIZE
180  K=INT(6*RND)
190  PRINT CUR(10,K*3) DBLE RED " "
195  PRINT CUR(10,23) FLSH K " Pistettä"
200  PRINT CUR(20,2) CYA "SEURAAVA SÄVELMÄ (RETURN)";
210  INPUT V$
220 NEXT N
```

7.1
```
10 REM OHJELMA          :KERTO1
20 REM OHJELMOIJA       : LAURI LUUPPI
30 REM PÄIVÄYS          : 1982-06-15
40 REM TIETOKONE        : ABC 800
200 PRINT CHR$(12)
220 PRINT "KAHDEN LUVUN KERTOLASKU"
240 PRINT "= = = = = = = = = = = = = = = = "
260 PRINT
280 INPUT "KIRJOITA ENSIMMÄINEN LUKU "A
300 INPUT "KIRJOITA TOINEN LUKU          "B
320 PRINT
340 PRINT "LUKUJEN TULO ON " A*B
360 PRINT CUR(18,0);
380 INPUT "HALUATKO JATKAA (KYLLÄ/EI) "V$
400 IF V$<>"EI" THEN GOTO 200
420 PRINT "AJO ON PÄÄTTYNYT"
430 END
```

7.2
```
10 REM OHJELMA          : KERTO2
270 ON ERROR GOTO 450
440 REM ------- VIRHEEN KÄSITTELIJÄ ----
450 PRINT " VIRHE !"
460 RESUME
```

7.3
```
4 REM OHJELMA          :L7H3
5 DIM A$(200)=40       : DIM B$(200)=20
10 WHILE A$(N)<>"LOPPU"
20  LET N=N+1
30  INPUT "NIMI/(LOPPU): "A$(N)
35  INPUT "PUHELIN:          "B$(N)
40 WEND
50 LET N=0
55 INPUT "KENEN PUHELINNUMERON HALUAT? "V$
60 WHILE A$(N)<>"LOPPU"
70  LET N=N+1
80  IF V$=A$(N) THEN PRINT A-ö-(N), B$(N)
90 WEND
100 INPUT "UUSI HAKU? (KYLLÄ/EI) "V$
110 IF V$="KYLLÄ" THEN GOTO 50
120 PRINT "AJO PÄÄTTYNYT"
```

7.4
```
100 REM L7H4
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "LASKELMA"
140 PRINT "= = ="
150 INPUT "PALKANKOROTUS/KK          (MK) : "Palkankorotus
160 INPUT "MARGINAALIVERO            (%)  : "Marginaalivero
170 INPUT "TYÖTUNTEJA/KK                  : "Työtunnit
180 PRINT
190 PRINT "PALKANKOROTUS/TUNTI       : ";
200 PRINT USING "# #.# #" (Palkankorotus-Marginaalivero*Palkankorotus/
    100)/Työtunnit;
210 PRINT " MARKKAA"
220 PRINT
```
Ratkaisut:

7.5
```
ERASE 100—180
LIST VIIVE.DEF
```

7.6
```
100 REM OHJELMA                    : L7H6
110 ON ERROR GOTO 110
120 REM ...
130 REM ...
140 CHAIN "VARASTO"
145 END
150 REM ------- VIRHEEN KÄSITTELIJÄ -------
160 PRINT
170 PRINT "TIEDOSTO EI OLE LEVYKKEELLÄ"
180 RESUME 145
```

7.8

```
100 REM OHJELMA                          : L7H8
110 REM OHJELMOIJA                       : MAIJA SUOMINEN
120 REM PÄIVÄYS                          : 1982-06-20
130 REM TIETOKONE                        : ABC 800
140 EXTEND
150 PRINT CHR$(12)
160 PRINT "LASKENTAOHJELMA"
170 PRINT
180 PRINT "VALITSE JOKIN SEURAAVISTA :"
190 PRINT
200 PRINT "1 YHTEENLASKU"
210 PRINT "2 VÄHENNYSLASKU"
220 PRINT "3 KERTOLASKU"
230 PRINT "4 JAKOLASKU"
240 PRINT "5 AJON LOPETUS"
250 PRINT
260 ON ERROR GOTO 690
270 INPUT "MINKÄ VAIHTOEHDON VALITSET ? "Vaihtoehto
280 PRINT
290 ON Vaihtoehto GOSUB 340,390,450,500,310
300 GOTO 150
310 REM ------- LOPETUS ----------
320 PRINT "AJO PÄÄTTYNYT"
330 END
340 REM ------- VAIHTOEHTO 1 ----------
350 GOSUB 550
360 PRINT "LUKUJEN " A " JA " B " SUMMA ON "A+B
370 GOSUB 650
380 RETURN
390 REM ------- VAIHTOEHTO 2 ----------
400 GOSUB 550
410 PRINT "LUKUJEN " A " JA " B " EROTUS ON " A—B
430 GOSUB 650
440 RETURN
450 REM ------- VAIHTOEHTO 3 ----------
460 GOSUB 550
470 PRINT "LUKUJEN " A " JA " B " TULO ON " A*B
480 GOSUB 650
490 RETURN
500 REM ------- VAIHTOEHTO 4 ----------
510 GOSUB 550
520 PRINT "LUKU " A " JAETTUNA LUVULLA " B " ON " A/B
530 GOSUB 650
540 RETURN
550 REM ------- SYÖTTÖ ----------
555 ON ERROR GOTO 720
560 PRINT CHR$(12)
570 PRINT "SYÖTTÖ: "
580 PRINT
590 INPUT "ANNA LUKU A: "A
600 INPUT "ANNA LUKU B: "B
610 PRINT
620 PRINT "TULOS:"
630 PRINT
640 RETURN
650 REM ------- TULOSTUS ----------
660 PRINT CUR(20,0);
```

```
670 INPUT "PALUU MENUUN (RETURN) "V$
680 RETURN
690 REM ------- VIRHERUTIINI 1 -------
700 PRINT " VIRHE !"
710 RESUME 150
720 REM ------- VIRHERUTIINI 2 -------
725 PRINT
730 RESUME
```

8.1

```
100 REM OHJELMA                          : L8H1
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "KIRJOITETAAN 25 TIETUETTA"
140 PRINT "TIEDOSTOON YRITYS1.DAT"
150 PRINT
160 PREPARE "YRITYS1.DAT" AS FILE 1
170 REM
180 FOR Tietuenumero=1 TO 25
190  REM
200  REM ------- SYÖTTÖ ----------
220  PRINT "SYÖTÄ NUMERO: " Tietuenumero
230  INPUT "TULOS:                "Tulos
240  REM
250  REM ------- TIEDOSTOON KIRJOITUS ----
260  PRINT#1, Tulos
290  REM ------------------------
300  PRINT
310 NEXT Tietuenumero
320 CLOSE 1
330 PRINT "AJO PÄÄTTYNYT"
340 END
```

8.2

```
100 REM OHJELMA                          : L8H2
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "LUETAAN PERÄKKÄISTIEDOSTOSTA 25 TIETUETTA,"
140 PRINT "JOISTA KUKIN SISÄLTÄÄ YHDEN LUVUN"
150 PRINT
180 PRINT
190 OPEN "YRITYS1.DAT" AS FILE 1
200 PRINT "LUVUT:"
210 PRINT
220 REM
230 FOR Tietuenumero=1 TO 25
240  REM
```

```
250  REM ------- TIEDOSTOSTA LUKU ----
260  INPUT #1, Tulos
270  REM
280  REM ------- TULOSTUS ----------
290  PRINT Tulos,
300  REM ------- SUMMAUS ----------
310  Summa=Summa+Tulos
320  REM ------------------------
330 NEXT Tietuenumero
340 CLOSE 1
350 PRINT : PRINT
360 PRINT "KESKIARVO ON : " Summa/25
370 PRINT
380 PRINT "AJO PÄÄTTYNYT"
390 END
```

8.3

```
100 REM OHJELMA                              : L8H3
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "LUETAAN PERÄKKÄISTIEDOSTOSTA 25 TIETUETTA,"
140 PRINT "JOISTA KUKIN SISÄLTÄÄ YHDEN LUVUN"
150 PRINT
160 REM ------- SYÖTTÖ ----------
170 INPUT "ANNA TIEDOSTON NIMI : "Tiedosto$
180 PRINT
190 OPEN Tiedosto$ AS FILE 1
200 PRINT "LUVUT:"
210 PRINT
220 REM
230 FOR Tietuenumero=1 TO 25
240  REM
250  REM ------- TIEDOSTOSTA LUKU ----
260  INPUT #1, Tulos
270  REM
280  REM ------- TULOSTUS ----------
290  PRINT Tulos,
300  REM ------- SUMMAUS ----------
310  Summa=Summa+Tulos
320  REM ------------------------
330 NEXT Tietuenumero
340 CLOSE 1
350 PRINT : PRINT
360 PRINT "KESKIARVO ON : " Summa/25
370 PRINT
380 PRINT "AJO PÄÄTTYNYT"
390 END
```

8.4

```
100 REM OHJELMA                              : L8H4
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "LUETAAN HALUTTU TIETUE"
140 PRINT "PERÄKKÄISTIEDOSTOSTA VARASTO2.DAT"
150 PRINT
160 REM ------- SYÖTTÖ ----------
170 INPUT "MINKÄ TUOTENUMERON HALUAT? "Vastaus
180 PRINT
190 OPEN "VARASTO2.DAT" AS FILE 1
```

```
200 PRINT
210 REM ------- TIEDOSTON LUKU -------
220 ON ERROR GOTO 390
230 INPUT #1, Tuotenumero
240 INPUT #1, Tuotenimi$
250 INPUT #1, Määrä
260 REM
270 REM ------- TULOSTUS ----------
280 IF Vastaus< >Tuotenumero THEN GOTO 330
290 PRINT "TUOTENRO:","TUOTENIMI:","MÄÄRÄ:"
300 PRINT Tuotenumero,
310 PRINT Tuotenimi$,
320 PRINT Määrä
330 GOTO 210
340 REM ------------------
350 CLOSE 1
360 PRINT
370 PRINT "AJO PÄÄTTYNYT"
380 END
390 REM ------- VIRHEEN KÄSITTELIJÄ -------
400 RESUME 350
```

8.5

```
100 REM OHJELMA                          : L8H5
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "LUETAAN 21 TIETUETTA"
140 PRINT "SUORASAANTITIEDOSTOSTA VARASTO3.DAT"
150 PRINT "JA TULOSTETAAN TILAUSLISTA"
160 PRINT
170 Kentänpituus1 = 36
180 Kentänpituus2 = 4
190 Tietueenpituus = Kentänpituus1 + Kentänpituus2
200 OPEN "VARASTO3.DAT" AS FILE 1
210 PRINT "TILAUSLISTA"
220 PRINT "= = = = = = ="
230 FOR Tietuenumero = 0 TO 20
240  REM
250  REM ------- TIEDOSTON LUKU -------
260  Tietueenpaikka = Tietuenumero * Tietueenpituus
270  POSIT #1, Tietueenpaikka
280  GET #1, Tuotenimi$ COUNT Kentänpituus1
290  GET #1, Määrä$ COUNT Kentänpituus 2
300  REM
310  REM ------- TULOSTUS ----------
320  Määrä = CVT$F(Määrä$)
330  IF Määrä > 10 THEN 360
340  PRINT Tuotenimi$
350  REM ------------------------
360 NEXT Tietuenumero
370 CLOSE 1
380 PRINT
390 PRINT "AJO PÄÄTTYNYT"
400 END
```

8.6

```
100 REM OHJELMA                                    :L8H6
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "ALUSTETAAN SUORASAANTITIEDOSTO, JOSSA ON"
140 PRINT "HALUTTU MÄÄRÄ ILMOITETUN MITTAISIA TIETUEITA"
150 PRINT
160 ON ERROR GOTO 330
170 REM ------- SYÖTTÖ ----------
180 INPUT "ANNA TIEDOSTON NIMI :                    "Tiedosto$
190 INPUT "ANNA TIETUEEN PITUUS (MERKKIEN LKM) :    "Tietueenpituus
200 INPUT "ANNA TIETUEIDEN LKM :                    "Tietuelkm
210 REM
220 REM ------- TIEDOSTON ALUSTUS ----
230 PREPARE Tiedosto$ AS FILE 1
240 FOR Tietuenumero=0 TO Tietuelkm—1
250  PUT #1, SPACE$(Tietueenpituus)
260 NEXT Tietuenumero
270 CLOSE 1
280 REM ------- LOPETUS ----------
290 PRINT
300 PRINT "TIEDOSTON " Tiedosto$
310 PRINT "ALUSTUS ON PÄÄTTYNYT"
320 END
330 REM ------- VIRHEEN KÄSITTELIJÄ ----
340 RESUME 170
```

8.7

```
100 REM OHJELMA                                    : L8H7
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "ALUSTETAAN SUORASAANTITIEDOSTO , JOSSA ON"
140 PRINT "HALUTTU MÄÄRÄ ILMOITETUN MITTAISIA TIETUEITA"
150 PRINT
160 ON ERROR GOTO 340
170 REM ------- SYÖTTÖ ----------
180 PRINT
190 INPUT "ANNA TIEDOSTON NIMI :                    "Tiedosto$
200 INPUT "ANNA TIETUEEN PITUUS (MERKKIEN LKM) :    "Tietueenpituus
210 INPUT "ANNA TIETUEIDEN LKM :                    "Tietuelkm
220 REM
230 REM ------- TIEDOSTON ALUSTUS -------
240 PREPARE Tiedosto$ AS FILE 1
250 FOR Tietuenumero=0 TO Tietuelkm—1
260  PUT #1, SPACE$(Tietueenpituus)
270 NEXT Tietuenumero
280 CLOSE 1
290 REM ------- LOPETUS ----------
300 PRINT
310 PRINT "TIEDOSTON " Tiedosto$
320 PRINT "ALUSTUS ON PÄÄTTYNYT"
330 END
340 REM ------- VIRHEEN KÄSITTELIJÄ ----
350 IF ERRCODE<>41 THEN RESUME 170
360 PRINT
370 PRINT "TIEDOSTO" Tiedosto$ "EI MAHDU LEVYLLE"
380 KILL Tiedosto$
390 RESUME 170
```

8.8

```
100 REM OHJELMA                          : L8H8
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "LUETAAN 21 TIETUETTA"
140 PRINT "SUORASAANTITIEDOSTOSTA VARASTO3.DAT"
150 PRINT "(Lopetus näppäimellä PF1)"
160 PRINT
170 Kentänpituus1 = 36
180 Kentänpituus2 = 4
190 Tietueenpituus = Kentänpituus1 + Kentänpituus2
200 OPEN "VARASTO3.DAT" AS FILE 1
210 ON ERROR GOTO 500
220 FOR Tietuenumero = 0 TO 20
230  REM
240  REM ------- Tiedoston lukeminen ----------
250  IF Tietuenumero < 0 THEN Tietuenumero = 0
260  Tietueenpaikka = Tietuenumero * Tietueenpituus
270  POSIT #1, Tietueenpaikka
280  GET #1, Tuotenimi$ COUNT Kentänpituus1
290  POSIT #1, Tietueenpaikka + Kentänpituus1
300  GET #1, Määrä$ COUNT Kentänpituus2
310  Määrä = CVT$F(Määrä$)
320  REM
330  REM ------- TULOSTUS ----------
340  PRINT
350  PRINT "TUOTENUMERO:              "Tietuenumero
360  PRINT
370  PRINT "TUOTENIMI:              "Tuotenimi$
380  PRINT "MÄÄRÄ VARASTOSSA:        "Määrä
390  PRINT CUR(20,0)
400  PRINT "PAINA RETURN UUTTA TIETUETTA VARTEN"
410  INPUT "(Edellinen tietue näppäimellä PF1)"Vastaus$
420  PRINT CHR$(12)
430  REM ------------------------
440 NEXT Tietuenumero
450 CLOSE 1
460 PRINT
470 PRINT "AJO PÄÄTTYNYT"
480 END
490 END
500 REM ------- VIRHEEN KÄSITTELIJÄ ----
510 IF ERRCODE <>53 THEN PRINT : RESUME
520 Buffer = SYS(6)
530 GET Tangent$
540 Pftangent = ASCII(Tangent$)—191
550 IF Pftangent > 2 THEN Pftangent = 1
560 ON Pftangent GOTO 580,600
570 REM
580 REM ------- PF1 ----------
590 RESUME 450
600 REM ------- PF2 ----------
610 Tietuenumero = Tietuenumero—1
620 PRINT CHR$(12)
630 RESUME 240
```

9.1
```
100 REM OHJELMAN NIMI: L9H1
110 EXTEND
120 PRINT CHR-◇-(12)
130 PRINT "TARKISTA ETTÄ KIRJOITIN ON KYTKETTY"
140 PRINT
150 INPUT "ANNA NIMESI:", Nimi-◇-
160 PREPARE "PR: " AS FILE 1
170 FOR Kierros=1 TO 30
180  PRINT #1, TAB(Kierros) Nimi-◇-
190 NEXT Kierros
200 END
```

9.2
```
100 REM OHJELMAN NIMI: L9H2
110 EXTEND
120 PRINT CHR-◇-(12)
130 PRINT "LUETAAN 5 TIETUETTA"
140 PRINT "PERÄKKÄISTIEDOSTOSTA VARASTO1.DAT"
145 PRINT "JA TULOSTUS PAPERILLE"
150 PRINT
160 OPEN "VARASTO1.DAT" AS FILE 1
165 PREPARE "PR:" AS FILE 2
170 REM
180 PRINT #2, "TUOTENRO:", "TUOTENIMI:", "MÄÄRÄ:"
190 PRINT #2,
200 FOR Tietuenumero=1 TO 5
210  REM
220  REM ------- TIEDOSTON LUKU -------
230  INPUT #1, Tuotenumero
240  INPUT #1, Tuotenimi-◇-
250  INPUT #1, Määrä
260  REM
270  REM ------- TULOSTUS ------------
280  PRINT #2, Tuotenumero,
290  PRINT #2, Tuotenimi-◇-,
300  PRINT #2, Määrä
310  REM ------------------------------
320  PRINT
330 NEXT Tietuenumero
340 CLOSE 1
350 PRINT "AJO PÄÄTTYNYT"
360 END
```

9.3
```
100 REM OHJELMAN NIMI: L9H3
272 Tuotenimi-◇- = Tuotenimi-◇- + SPACE-◇-(36)
274 Tuotenimi-◇- = LEFT-◇-(Tuotenimi-◇-,36)
```

9.4

```
100 REM OHJELMAN NIMI: L9H4
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "ANNA MIELIVALTAINEN RUOKALISTAN NIMI"
140 PRINT
150 INPUT " "Rivi$
160 PRINT
170 PRINT "OIKEALTA VASEMMALLE LUETTAVA RIVI"
180 PRINT
190 Määrä = LEN(Rivi$)
200 FOR Merkki = Määrä TO 1 STEP -1
210  PRINT MID$(Rivi$,Merkki,1);
220 NEXT Merkki
230 PRINT : PRINT
240 END
```

9.5

```
100 REM OHJELMAN NIMI: L9H5
110 PRINT CHR$(12)
120 GET S$ : PRINT CHR$(ASCII(S$) - 1); : GOTO 120
```

9.6

```
100 REM OHJELMAN NIMI: L9H6
110 EXTEND
120 PRINT CHR$(12)
130 INPUT "ANNA RIVIN MERKKIEN LUKUMÄÄRÄ "Määrä
140 PRINT
150 PRINT "ANNA TOIVOTTU OTSIKKO:"
160 INPUT " "Otsikko$
170 PRINT CHR$(12)
180 Tabpaikka = INT((Määrä—LEN(Otsikko$))/2)
190 IF Tabpaikka < 1 THEN Tabpaikka = 1
200 PRINT TAB(Tabpaikka) Otsikko$
210 PRINT TAB(Tabpaikka) STRING$(LEN(Otsikko$),61)
220 PRINT CUR(20,0);
230 END
```

9.7

```
100 REM OHJELMAN NIMI: L9H7
360 PRINT "OHJELMAN VIRHE!" : PRINT CHR$(7);
```

9.8

```
100 REM OHJELMAN NIMI: L9H8
110 EXTEND
120 PRINT CHR-◊-(12)
130 PRINT "21 TIETUEEN LUKU"
140 PRINT "SUORASAANTITIEDOSTOSTA VARASTO3.DAT JA"
150 PRINT "AAKKOSTETUN LUETTELON TULOSTUS KIRJOITTIMELLA"
160 PRINT
170 Kentänpituus1 = 36
180 Kentänpituus2 = 4
190 DIM Nimike-◊-(20)
200 DIM Määrä(20)
210 Tietueenpituus = Kentänpituus1 + Kentänpituus2
220 OPEN "VARASTO3.DAT" AS FILE 1
230 REM
240 REM ------- TIEDOSTON LUKU ------
250 REM
260 FOR Tietuenumero = 0 TO 20
270  Tietueenpaikka = Tietuenumero * Tietueenpituus
280  POSIT #1, Tietueenpaikka
290  GET #1, Nimike-◊-(Tietuenumero) COUNT Kentänpituus1
300  POSIT #1, Tietueenpaikka + Kentänpituus1
310  GET #1, Määrä-◊- COUNT Kentänpituus2
320  Määrä(Tietuenumero) = CVT-◊-F(Määrä-◊-)
330  REM
340 NEXT Tietuenumero
350 CLOSE 1
360 REM ------- LAJITTELU -------
370 REM
380 FOR Indeksia = 0 TO 19
390  FOR Indeksib = Indeksia + 1 TO 20
400     IF Nimike-◊-(Indeksia) < Nimike-◊-(Indeksib) THEN 470
410     Apumuuttuja-◊- = Nimike-◊-(Indeksia)
420     Nimike-◊-(Indeksia) = Nimike-◊-(Indeksib)
430     Nimike-◊-(Indeksib) = Apumuuttuja-◊-
440     Apumuuttuja = Määrä(Indeksia)
450     Määrä(Indeksia) = Määrä(Indeksib)
460     Määrä(Indeksib) = Apumuuttuja
470  NEXT Indeksib
480 NEXT Indeksia
490 REM
500 REM ------- TULOSTUS KIRJOITTIMELLA -------
510 OPEN "PR:" AS FILE 2
520 PRINT #2,
530 PRINT #2, "TUOTELUETTELO AAKKOSJÄRJESTYKSESSÄ"
540 PRINT #2,
550 PRINT #2, "TUOTE" TAB(39) "MÄÄRÄ"
560 PRINT #2,
570 FOR Numero = 0 TO 20
580  PRINT #2, Nimike-◊-(Numero);
590  PRINT #2, USING "# # # # # # #" Määrä(Numero)
600 NEXT Numero
610 CLOSE 2
620 PRINT
630 PRINT "AJO SUORITETTU"
640 END
```

9.9

```
100 REM OHJELMAN NIMI: L9H9
110 EXTEND
120 PRINT CHR-ϕ-(12)
130 DIM Nimi-ϕ-(100)
140 PRINT "NIMIEN LAJITTELU (ENINTÄÄN 100 NIMEÄ)"
150 PRINT "= = = = = = = = = = = = = = = = = = = = ="
160 PRINT
170 REM ------- SYÖTTÖ -------
180 REM
190 INPUT "LAJITELTAVIEN NIMIEN LUKUMÄÄRÄ "Määrä
200 PRINT
210 FOR Numero=1 TO Määrä
220  PRINT "ANNA NIMI " Numero;
230  INPUT " "Nimi-ϕ-(Numero)
240 NEXT Numero
250 REM ------- LAJITTELU -------
260 REM
270 FOR Apaikka=1 TO Määrä -1
280  FOR Bpaikka=Apaikka+1 TO Määrä
290    IF Nimi-ϕ-(Apaikka)<Nimi-ϕ-(Bpaikka) THEN GOTO 330
300    Apumuuttuja-ϕ- =Nimi-ϕ-(Apaikka)
310    Nimi-ϕ-(Apaikka)=Nimi-ϕ-(Bpaikka)
320    Nimi-ϕ-(Bpaikka)=Apumuuttuja-ϕ-
330  NEXT Bpaikka
340 NEXT Apaikka
350 GOSUB 420 : REM TULOSTUS
360 REM ------- LOPETUS -------
370 REM
380 PRINT : PRINT : PRINT
390 PRINT "LAJITTELU VALMIS JA AJO LOPPUNUT"
400 PRINT : PRINT
410 END
420 REM ----- TULOSTUS.SUB -----
430 PRINT
440 OPEN "PR:" AS FILE 1
450 PRINT #1, "AAKKOSTETTU LUETTELO"
460 PRINT #1, "= = = = = = = = = = = = = ="
470 PRINT #1,
480 FOR Numero=1 TO Määrä
490  PRINT #1, Nimi-ϕ-(Numero)
500 NEXT Numero
510 RETURN
```

9.10

```
100 REM OHJELMAN NIMI: L9H10
110 EXTEND
120 LET S=10000
130 PRINT CHR$(12)
140 PRINT "P I K A O T T O"
150 PRINT "= = = = = = = = ="
160 PRINT CUR(4,0) "TERVETULOA !"
170 INPUT "ANNA SALAINEN KOODI : "S$
180 GOSUB 340
190 IF S$<>"RAHA" THEN GOTO 170
200 INPUT "OTTO? "B
210 IF B>S THEN LET B=S : PRINT CHR$(7)
220 LET S = S—B
230 PRINT CHR$(12)
240 PRINT "P I K A O T T O"
250 PRINT "= = = = = = = = ="
260 PRINT CUR(4,0) "PÄIVÄYS: AIKA:"
270 PRINT TIME$
280 PRINT CUR(7,0) "OTETTU SUMMA: "B
290 PRINT "TILILLÄ JÄLJELLÄ: " S
300 PRINT CUR(11,0) "TERVETULOA UUDELLEEN!"
310 FOR N=1 TO 15000
320 NEXT N
330 GOTO 130
340 REM ------- VALINTA.SUB -------
350 FOR Merkki=1 TO LEN(S$)
360  Merkki$=MID$(S$,Merkki,1)
370  Kod=ASCII(Merkki$)
380  IF Kod>95 THEN Kod=Kod—32
390  MID$(S$,Merkki,1)=CHR$(Kod)
400 NEXT Merkki
410 RETURN
```

10.1

```
100 REM OHJELMAN NIMI: L10H1
110 DOUBLE
120 EXTEND
130 PRINT CHR$(12)
140 PRINT "FUNKTION Y = SIN(X) ARVOTAULUKKO"
150 PRINT "= = = = = = = = = = = = = = = = = = = = ="
160 PRINT
170 PRINT TAB(7) "X" TAB(22) "Y=SIN(X)"
180 PRINT
190 FOR X=0 TO 2*PI STEP PI/7
200  PRINT USING "# #.# # # # # # # # # #" X TAB(20) SIN(X)
210 NEXT X
220 PRINT
230 END
```

10.2
```
100 REM OHJELMAN NIMI: L10H2
110 EXTEND
120 DOUBLE
130 PRINT CHR-ϕ-(12)
140 PRINT "ESIMERKKI 'DIGITSIN' KÄYTÖSTÄ"
150 PRINT "= = = = = = = = = = = = = = = = = ="
160 PRINT
170 FOR Määrä = 1 TO 16
180  DIGITS Määrä
190  PRINT "PI "Määrä TAB(12) "NUMEROLLA " PI
200 NEXT Määrä
210 PRINT
220 END
```

10.3
```
100 REM OHJELMAN NIMI: L10H3
110 DOUBLE
120 EXTEND
130 PRINT CHR-ϕ-(12)
140 PRINT "EKSPONENTTI" TAB(27) "2**EKSPONENTTI"
150 PRINT
160 FOR Eksponentti = 1 TO 50
170  PRINT CUR(5,0);
180  PRINT USING "# # # #" Eksponentti;
190  PRINT TAB(20);
200  PRINT USING "# # # # # # # # # # # # # # # # # #" 2üEksponentti
210  PRINT CUR(15,0) "PAINA RETURN KUN HALUAT JATKAA ";
220  GET Vastaus-ϕ-
230 NEXT Eksponentti
240 END
```

10.4
```
100 REM OHJELMAN NIMI: L10H4
110 EXTEND
120 DOUBLE
130 PRINT CHR-ϕ-(12)
140 INPUT "ANNA PÄÄOMA (MK)", Pääoma
150 INPUT "ANNA KORKO (%)", Korkoprosentti
160 INPUT "ANNA AIKA (VUOSISSA)", Vuosia
170 PRINT
180 PRINT "PÄÄOMA ANNETUN AJAN KULUTTUA"
190 PRINT "ON: ";
200 PRINT USING "# # # # # # # # # # # # # # # # # #.# #" FNPääoma (Pääoma,
Korkoprosentti,Vuosia);
210 PRINT " MK"
220 PRINT
230 END
240 DEF FNPääoma (Pääoma,Korkoprosentti,Vuosia) = Pääoma*
(1 + Korkoprosentti/100)üVuosia
```

10.5

```
100 REM OHJELMAN NIMI: L10H5
110 DOUBLE
120 PRINT CHR-ö-(12)
130 PRINT "FUNKTIOLLE Y=COS(X), 0 <= X <= 2*PI"
140 PRINT "LASKETAAN 50 ARVOA"
150 PRINT
160 PRINT "LUKUPARIT (X, Y) TALLENNETAAN TIEDOSTOON"
170 PRINT " 'TIEDCOS.DAT' "
180 PRINT
190 PRINT "ODOTA!"
200 PREPARE "TIEDCOS.DAT" AS FILE 1
210 FOR X=0 TO 2*PI STEP PI/50
220  PRINT #1,X
230  Y=COS(X)
240  PRINT #1,Y
250 NEXT X
260 CLOSE 1
270 PRINT
280 PRINT "TIEDOSTO VALMIS"
290 END
```

10.6

```
100 REM OHJELMAN NIMI: L10H6
110 EXTEND
120 PRINT CHR-ö-(12)
130 PRINT "Y:N PIENIMMÄN ARVON LASKEMINEN"
140 PRINT "LUKUPAREISTA (X, Y), JOTKA ON TALLENNETTU"
150 PRINT "SEURAAVASSA ANNETTAVAAN PERÄKKÄISTIEDOSTOON"
160 PRINT
170 PRINT "ANNA TIEDOSTON NIMI"
180 INPUT "(ESIM 'TIEDCOS.DAT')", Tiedosto-ö-
190 PRINT
200 PRINT "PIENIN ARVO ON " FNPieninarvo(Tiedosto-ö-)
210 PRINT
220 END
230 REM -----PIENINARVO.DEF------
240 DEF FNPieninarvo(Tiedosto-ö-) LOCAL Xtest,Ytest,Zmin
250  Ymin=0
260  OPEN Tiedosto-ö- AS FILE 1
270  ON ERROR GOTO 340
280  INPUT #1,Xtest
290  INPUT #1,Ytest
300  IF Ytest<Ymin THEN Ymin=Ytest
310  GOTO 280
320  CLOSE 1
330  RETURN Ymin
340  RESUME 320
350  FNEND
```

10.7
```
100 REM OHJELMAN NIMI: L10H7
110 EXTEND
120 PRINT CHR-⌖-(12)
130 PRINT "TILASTO-OHJELMA"
140 PRINT "= = = = = = = = = = = = = = = = ="
150 DIM Luku(15)
160 PRINT CUR(4,0) "ANNA 15 LUKUA:"
170 PRINT
180 ON ERROR GOTO 410
190 FOR Numero=1 TO 15
200  PRINT "LUKU NO " Numero;
210  INPUT "  = "Luku(Numero)
220  Summa=Summa+Luku(Numero)
230 NEXT Numero
240 FOR Apaikka=1 TO 15
250  FOR Bpaikka=Apaikka+1 TO 15
260    IF Luku(Apaikka)<Luku(Bpaikka) THEN GOTO 300
270    Apumuuttuja=Luku(Apaikka)
280    Luku(Apaikka)=Luku(Bpaikka)
290    Luku(Bpaikka)=Apumuuttuja
300  NEXT Bpaikka
310 NEXT Apaikka
320 PRINT CHR-⌖-(12)
330 PRINT "TULOS"
340 PRINT "= = = =       ="
350 PRINT CUR(5,0) "KESKIARVO      = " Summa/15
360 PRINT "MEDIAANI             = " Luku(8)
370 PRINT "PIENIN ARVO          = " Luku(1)
380 PRINT "SUURIN ARVO          = " Luku(15)
390 PRINT CUR(20,0) "AJO SELVÄ!"
400 END
410 PRINT "VIRHE !"
420 RESUME 200
```

10.8 —

11.1
```
100 REM OHJELMA                          : L11H1
110 EXTEND
120 PRINT CHR$(12)
125 DIM Pylväskorkeus(30)
130 PRINT "KOKONAISLUVUT ESITETÄÄN PYLVÄSDIAGRAMMINA"
140 PRINT "= = = = = = = = = = = = = = = = = = = = = = = = = = = = = ="
150 PRINT
160 PRINT "ANNA 30 KOKONAISLUKUA (PYLVÄÄN KORKEUTTA)"
170 PRINT "VÄLILLÄ 0 < KORKEUS < 100"
180 PRINT
190 REM ------- SYÖTTÖ ----------
200 FOR Indeksi=1 TO 30
210  PRINT "ANNA PYLVÄÄN KORKEUS " Indeksi;
220  INPUT " ",Pylväskorkeus(Indeksi)
230 NEXT Indeksi
240 REM ------- PIIRRÄ DIAGRAMMI -------
```

```
250 PRINT CHR$(12);
260 FOR Indeksi=1 TO 30
270  FOR Pylväs=1 TO Pylväskorkeus(Indeksi)
280    PRINT CUR(20—Pylväs/5,5+Indeksi); "■";
290  NEXT Pylväs
300 NEXT Indeksi
310 REM
320 REM ------- TULOSTA ASTEIKKO ----
330 PRINT CUR(0,0) "100—"
340 PRINT CUR(10,0) " 50—"
350 PRINT CUR(20,0);" 0 "
370 PRINT
380 END
```

11.2

```
100 REM OHJELMA                                    : L11H2
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "KOKONAISLUVUT ESITETÄÄN PYLVÄSDIAGRAMMINA"
140 PRINT "= = = = = = = = = = = = = = = = = = = = = = = = = ="
150 PRINT
160 PRINT "ANNA 7 KOKONAISLUKUA (PYLVÄÄN KORKEUTTA)"
170 PRINT "VÄLILLÄ 0 < KORKEUS < 20"
180 PRINT
190 REM ------- SYÖTTÖ ----------
200 FOR Indeksi=1 TO 7
210  PRINT "ANNA PYLVÄÄN KORKEUS " Indeksi;
220  INPUT " ",Pylväskorkeus(Indeksi)
230 NEXT Indeksi
240 REM ------- PIIRRÄ DIAGRAMMI -------
250 PRINT FNSinkel$;
260 FOR Indeksi=1 TO 7
270  FOR Pylväs=1 TO Pylväskorkeus(Indeksi)
280    PRINT Cur(20—Pylväs,2+Indeksi*5);"
290  NEXT Pylväs
300 NEXT Indeksi
310 REM
320 REM ------- TULOSTA ASTEIKKO ----
330 PRINT CUR(0,2) RED "20—" GBLU
340 PRINT CUR(10,2) RED "10—" GBLU
350 PRINT CUR(20,2);
360 PRINT RED "   1   2   3   4   5   6   7"
370 PRINT
380 END
390 REM ------- SINKEL.DEF ----------
400 DEF FNSinkel$ LOCAL Rivi
410  PRINT CHR$(12)
420  Rivi=0
430  WHILE Rivi<23
440    PRINT CUR(Rivi,0) YEL NWBG GBLU;
450    Rivi=Rivi+1
460  WEND
470  RETURN Eimitään$
480 FNEND
```

11.3

```
100 REM OHJELMA                          : L11H3
110 EXTEND
120 PRINT CHR$(12)
130 PRINT "LUVUT ESITETÄÄN PYLVÄSDIAGRAMMINA"
140 PRINT "= = = = = = = = = = = = = = = = = = = = = ="
150 PRINT
160 PRINT "ANNA 7 KOKONAISLUKUA (PYLVÄÄN KORKEUTTA)"
170 PRINT "VÄLILLÄ 0 < KORKEUS < 200"
180 PRINT
190 REM ------- SYÖTTÖ ----------
200 FOR Indeksi=1 TO 7
210  PRINT "ANNA PYLVÄÄN KORKEUS " Indeksi;
220  INPUT " ",Pylväskorkeus(Indeksi)
230  IF Pylväskorkeus(Indeksi)>199 THEN Pylväskorkeus(Indeksi)=199
240 NEXT Indeksi
250 REM ------- PIIRRÄ DIAGRAMMI -------
260 PRINT FNTyhjennä$
270 FGCTL 39
280 FOR Indeksi=1 TO 7
290  IF Pylväskorkeus(Indeksi)=0 THEN GOTO 320
300  FGPOINT 30*Indeksi,40,2
310  FGFILL 30*Indeksi+15,Pylväskorkeus(Indeksi)+40
320 NEXT Indeksi
330 REM
340 REM ------- TULOSTA ASTEIKKO ----
350 PRINT CUR(0,0) "200"
360 PRINT CUR(10,0) "100"
370 PRINT CUR(20,0) "   1   2   3   4   5   6   7"
380 END
390 REM ------- TYHJENNÄ.DEF -------
400 DEF FNTyhjennä$
410  PRINT CHR$(12)
420  FGCTL 0
430  FGPOINT 0,0,0
440  FGFILL 239,239
450  RETURN Eimitään$
460 FNEND
```

11.4

```
230 PRINT CUR(Rivi,0) RED NWBG GYEL;
```

11.5

```
100 REM OHJELMA                                  : L11H5
110 EXTEND
120 PRINT FNTyhjennä$;
130 FGCTL 8 : REM VÄRIYHDISTELMÄ 8
140 PRINT FNAkselit$
150 REM
160 REM --- LASKE JA PIIRRÄ ----------
170 Xskaalauskerroin=35 : Yskaalauskerroin=80
180 Xpiste=0 : Ypiste=100
190 FOR X=0 TO 2*PI STEP PI/100
200  FGPOINT Xskaalauskerroin*X,Ypiste+Yskaalauskerroin*SIN(X),2
210  FGPOINT Xskaalauskerroin*X,Ypiste+Yskaalauskerroin*COS(X),3
220 NEXT X
230 REM --- KIRJOITA OTSIKKO JA LOPETA --
240 PRINT CUR(3,10) YEL DBLE "Y1=SIN(X)," MAG "Y2=COS(X)"
250 PRINT CUR(6,13) DBLE RED "0 < X < 2*PI"
260 PRINT CUR(22,0)
270 END
280 REM ------- TYHJENNÄ.DEF -------
290 DEF FNTyhjennä$
300  PRINT CHR$(12)
310  FGCTL 0
320  FGPOINT 0,0,0
330  FGFILL 239,239
340  RETURN Eimitään$
350 FNEND
360 REM ------- AKSELIT.DEF -------
370 DEF FNAkselit$
380  FGPOINT 0,100,1
390  FGLINE 239,100
400  FGPOINT 0,20
410  FGLINE 0,200
420  FGPOINT 0,100 : REM ORIGO
430  RETURN Eimitään$
440 FNEND
```

13

Asia-hakemisto ja viite-kirjallisuus

SIAHAKEMISTO

kkosjärjestys 161
kkosnumeerinen merkki 78
ohjelma 113
kuteksti 31
ɔitusosa 105
ıstus 95
umuuttuja 163
cii-aritmetiikka 174
cii-arvo 158
cii-koodi 158
ırinkomerkki 29

ısic 22

ıtakuvaukset 107
ıtatiedostot 125
simaalimuoto 59
mensiointi 106
skette 90

itointi 55
litointinappulat 55
ıdollinen hyppy 34
ıdoton hyppy 33
ıto 34
ısisijaisuus 175
ity s 73

ɔppy disk 90
nktionäppäimet 13, 133
nktiot 113

aafinen merkki 78
afiikan ohjausmerkki 190
afiikkatila 79, 191

haku 42
hieno grafiikka 189

ikuinen silmukka 33
indeksi 37
indeksoitu muuttuja 40
integroitu piiri 11

järjestelmälevy 95
järjestelmäohjelma 95

kaiutin 158
kaksinkertainen teksti 76
kalvolevy 90
karkea grafiikka 189
kasettinauhuri 89
kello 65
kenttä 128
keskusmuisti 95
kirjaintila 79
kirjoitin 153
kirjoituspää 152
kohdistimen sijoitus 53
kohdistin 11
kokonaislukumuuttuja 172
komento 21
kommentti 105
konvertoiminen 142
kopiointinäppäin 55
kuvamuisti 195
kuvaruudun rivi 56
kuvaruudun tyhjentäminen 52
kuvaruutu 10
käsky 22
käskyosa 107

lajittelu 161
lause 22
lerppulevy 90
levyasema 89
levyke 90
liukulukumuuttuja 172
logaritmi 175
looginen loppu 106
loop 33
lukumuuttuja 39, 51

magneettinen laite 89
menu 112
menu-tekniikka 112
merkitsevä numero 171
merkkijonojen vertailu 161
merkkijonomuuttuja 29
merkkijonon pituus 155
merkkimuunnos 158
mikrotietokone 9
muokkausosa 107
muuttuja 29
muuttujakuvaus 107
muuttujanimi 110
muuttujatyyppi 172

noppapeli 61
numeerinen merkkijono 172
numeerinen näppäin 13
numeromerkki 127

ohjausmerkki 73
ohjelma 22
ohjelma-ajon keskeytys 33
ohjelman lopetus 107
ohjelman nimi 91, 107
ohjelman rakenne 107
ohjelmarivi 21, 22
ohjelmarivin muuttaminen 56
omat funktiot 177
osajono 160
osoite 92

paikallinen muuttuja 179
pankkiautomaatti 64
perusfunktiot 175
peräkkäistiedosto 125
poistonäppäin 18, 55
prosessi 80
päivitys 144

rivinumero 22

sanastin 154
sanojenkäsittely 154
sanojenkäsittelyohjelma 154
sarake 38
satunnaisluku 58
sektori 95
sektoridiagrammi 200
silmukka 33
standardivastaus 158
suorasaantitiedosto 135
suoritusaika 172
suurennettu teksti 76
syöttö 31

taustaväri 77
tekstinkäsittelyohjelma 154
tiedoston kopioiminen 146
tiedoston loppumerkki 125
tiedostonumero 128
tiedostotyyppi 93
tietolevy 90
tietosuojalaki 43
tietue 128
trigonometrinen funktio 175
tulostuksen muotoilu 171
tunnistusinformaatio 107
tv-monitori 10

ulkoinen muisti 89
ura 95

ıroitusmerkki 158
ivytys 62
lkkuva teksti 78
rheenkäsittely 108
rheilmoitus 93, 109
rhekoodi 134
ıokaavio 35
ıri 73
.rivalinta 73
.rivalintakäsky 195
:riyhdistelmä 197

ıteiset muuttujat 95
:sikön nimi 94
ɛinen tietokonejärjestelmä 10

nimerkki 158

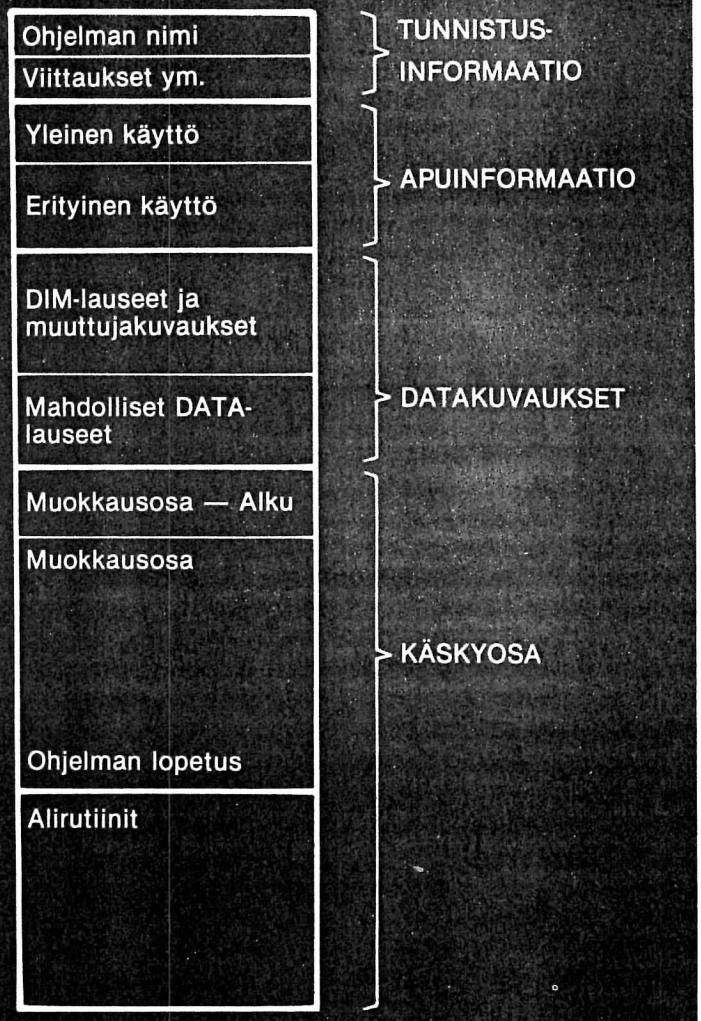
iitekirjallisuus

ABC 800 ja FACIT DTC käyttöohjeet
BASIC II -manuaali ABC 800- ja FACIT DTC -tietokoneille
Levymuistin käyttöohje
Kasettimuistin käyttöohje
BASIC-kielen ABC
Anders Andersson, Arne Kullbjer, Jan Lundgren,
Sören Thornell

ABC 800®

OHJELMOINTIKORTTI

OHJELMAN RAKENNE



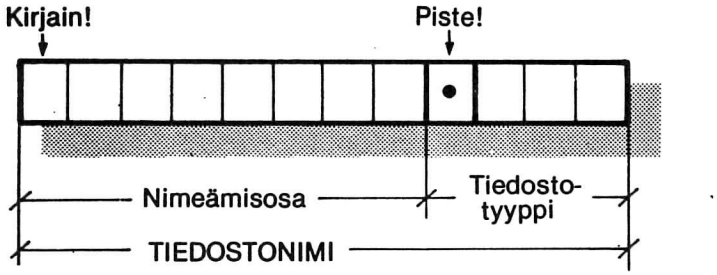
Luxor Data

MUUTTUJAMERKINNÄT

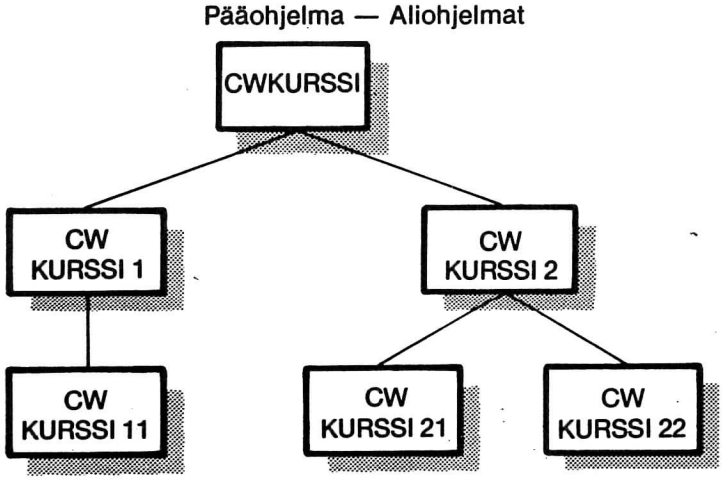
MERKINTÄ	DATALAJI	MUUTTUJA-TYYPPI
A—H	vakiot/syöttötiedot	liukuluku
A%—H%	vakiot/syöttötiedot	kokonaisluku
A¤—H¤	vakiot/syöttötiedot	jonomuuttuja
I—N	indeksit/silmukkamuuttujat	liukuluku
I%—N%	indeksit/silmukkamuuttujat	kokonaisluku
O	tulee välttää, koska	liukuluku
O%	kirjain voidaan sekoittaa	kokonaisluku
O¤	nollaan	jonomuuttuja
P—U	välitulosten tallettamiseen	liukuluku
P%—U%	käytettäviä muuttujia, nk.	kokonaisluku
P¤—U¤	väliaikaismuuttujia	jonomuuttuja
V—Z	tulosmuuttuja	liukuluku
V%—Z%	tulosmuuttuja	kokonaisluku
V¤—Z¤	tulosmuuttuja	jonomuuttuja

Ylläolevissa muuttujamerkinnöissä voidaan kaikki kirjaimet korvata kirjaimen ja numeron (0—9) yhdistelmällä, esim. A8, B3%, C5¤. Muuttujanimissä voidaan käyttää valinnaista määrää merkkejä, jos ohjelman alussa on käsky EXTEND (tällöin vaaditaan välilyönnin oikeata käyttöä ohjelmaa syötettäessä).

NIMEÄMISSÄÄNTÖJÄ



Yhdessä nimen kanssa annetaan toisinaan tiedoston nykyinen yksikkö, esim. DR0:, DR1:, CAS:



CHAIN "... ... ..."

Siirtää ilmoitetun ohjelman kasetilta tai levyltä ABC 800:aan ja aloittaa ohjelman ajon. Käskyllä on sama toiminta kuin käskyllä RUN ... ... Ks. myös COMMON ...

```
20 CHAIN "CAS:LISTA.007"
60 CHAIN "OHJELMA2"
90 CHAIN A¤
```

CLOSE ...

Sulkee ilmoitetun tiedoston (tiedostot), ts. lopettaa tiedoston lukemisen tai kirjoittamisen. Jos tiedosto on avattu kirjoitusta varten, siihen kirjoitetaan tiedostonloppumismerkki. Jos ilmoitettu tiedosto on talletettu kasetille, nauhurin moottori pysähtyy. Mikäli tiedostonumeroa ei ilmoiteta, suljetaan kaikki avatut tiedostot.

```
50 CLOSE 4
60 CLOSE 1, 4
70 CLOSE
```

COMMON ...

Ilmoitetut muuttujat eivät nollaudu, kun käsky CHAIN "... ... ..." toteutetaan. Ilmoitetut muuttujat voidaan myös dimensioida. Vrt. DIM ... Ohjelmarivit, jotka sisältävät COMMON-ilmoituksia, on sijoitettava ohjelman alkuun.

```
10 COMMON X, Y, Z
20 COMMON X¤(30) = 100, Y¤(20) = 200
```

DATA ...

Tallettaa muuttuja-arvot, jotka luetaan sitten READ-lauseella. Merkkijonot, jotka sisältävät pilkkuja, tulee ympäröidä sitaatein.

```
40 DATA 13, 2, 6, RIITTA
50 DATA "Allekirjoittanut, PELLE PELKONEN"
```

DEF FN ... (...) = ...

Määrittelee oman funktion yhdellä rivillä.

```
70 DEF FNY(R) = PI*R*R
80 DEF FNZ(B,H) = (B*H)/2
90 DEF FNX¤(S¤) = LEFT¤(S¤,LEN(S¤)—2)
```

DEF FN ... (...) LOCAL ...

Määrittelee oman monirivisen funktion. Ilmoituksella LOCAL ... määritellään paikallismuuttujat. LOCAL ... voidaan jättää pois. Funktion arvo palautetaan käskyllä RETURN ... FNEND-käsky lopettaa määrittelyn.

```
100 DEF FNY(R) LOCAL (S)
200 S = R/180
300 RETURN S*R
400 FNEND
```

DIGITS ...

Rajoittaa esitettävien numeroiden määrän tulostettaessa muuttujan arvo käskyllä PRINT ... Ei vaikuta laskentatarkkuuteen.

```
10 DIGITS 4
70 DIGITS N
```

DIM ...

Varaa tilaa kenttämuuttujille. Pienin indeksi on normaalisti 0, mutta se voidaan muuttaa ykköseksi käskyllä OPTION BASE 1.

```
10 DIM P(20)
```

Varaa tilaa 21 liukulukumuuttujalle P(0) — P(20).

```
20 DIM P¤ = 1000
```

Varaa tilaa merkkijonolle, jossa on enintään 1000 merkkiä.

```
30 DIM P¤(20) = 10
```

Varaa tilaa 21 jonomuuttujalle, joissa kussakin enintään 10 merkkiä.

```
40 DIM P%(10,20,5)
```

Varaa tilaa matriisille, joka käsittää 11 × 21 × 5 kokonaislukumuuttujaa.

```
50 DIM P¤(10,20) = 40
```

Varaa tilaa merkkijonomatriisille, jossa on 11 riviä ja 21 saraketta ja joissa kukin alkio käsittää enintään 40 merkkiä.

DOUBLE

Aiheuttaa liukulukujen esittämisen 16 numeron tarkkuudella normaalin 7 numeron sijaan. Käskyn tulee olla ensimmäisenä ohjelmassa. Ks. myös SINGLE.

END

Lopettaa ohjelman ajon ja sulkee kaikki tiedostot.

EXTEND

Sallii pitkät muuttujanimet. Vrt. NO EXTEND.

```
10 EXTEND
20 Bruttohinta = Nettohinta * 1.25
```

FLOAT

Aiheuttaa sen, että muuttuja-arvoja ja vakioita käsitellään normaalisti liukulukuina. Kokonaislukumuuttujat ja kokonaislukuvakiot merkitään %-merkillä. FLOAT sijoitetaan ensimmäiseksi ohjelmaan. Vrt. INTEGER.

```
10 FLOAT
20 A = 2.567 : B% = 9%
```

FNEND

Lopettaa oman moniarvoisen funktion määrittelyn. Ks. DEF FN ... (...)

FOR ... TO ... STEP ...
NEXT

Toistaa ohjelmanosan niin monta kertaa kuin silmukkamuuttujan alku ja loppuarvo sekä askelarvo määräävät. Jos askelasrvo jätetään pois, se asetetaan arvoon 1.

```
20 FOR K = 1 TO 20 STEP 2
40 NEXT K
60 FOR L% = A% * 3% TO B%
90 NEXT L%
```

KÄSKYT (jatkoa)

GET ...

Odottaa näppäimenpainallusta. Tallettaa vastaanotetun merkin ilmoitettuun jonomuuttujaan kirjoittamatta merkkiä kuvaruutuun. Kaikki merkit hyväksytään.

```
20 GET S7¤
```

GET # ... , ... COUNT ...

Siirtää dataa ilmoitetusta suorasaantitiedostosta haluuttuun jonomuuttujaan. Siirrettävien merkkien määrä ilmoitetaan COUNT-sanan jäljessä. Jos COUNT ... jätetään pois, siirretään vain yksi merkki. Tiedosto-osoitin päivittyy automaattisesti. Vrt. POSIT (...).

```
90 GET #1, S¤ COUNT 2000
```

GOSUB ...

Aiheuttaa hypyn alirutiiniin ilmoitetulla rivillä. Vrt. RETURN.

```
80 GOUSB 2000
```

GOTO ...

Aiheuttaa hypyn ilmoitetulle riville.

```
60 GOTO 10
```

IF ... THEN ... ELSE ...

Jos IF-sanan jälkeinen ilmaus on tosi (=0), silloin suoritetaan se, mitä THEN- ja ELSE-sanojen väliin on kirjoitettu. Muussa tapaukessa suoritetaan se, mitä on kirjoitettu ELSE-sanan jälkeen. ELSE ... voidaan jättää pois.

```
40 IF A=0 THEN GOTO 60 ELSE 10
```

(GOTO voidaan jättää pois THEN- ja ELSE-sanojen jäljessä)

INPUT "..." ...

Kirjoittaa annetun alkutekstin kuvaruutuun ja odottaa sen jälkeen datan syöttöä näppäimistön kautta. Alkuteksti voidaan jättää pois.

```
30 INPUT "Anna nimesi " S¤
90 INPUT P
```

INPUT # ... , ...

Siirtää dataa ilmoitetusta tiedostosta ilmoitettuun muuttujaan.

```
50 INPUT #2,R
60 INPUT #F,R¤
```

INPUT LINE ...

Siirtää näppäimistöllä syötetyn rivin ilmoitettuun jonomuuttujaan. Kaikki merkit hyväksytään. Jonomuuttujan kaksi viimeistä merkkiä ovat aina CR (vaunupalautus) ja LF (rivinsiirto). Syötetyt merkit CR:ää ja LF:ää lukuunottamatta tulostetaan kuvaruutuun.

```
90 INPUT LINE S¤
```

INPUT LINE # ... , ...

Siirtää dataa ilmoitetusta tiedostosta ilmoitettuun jonomuuttujaan. Kaikki merkit, välimerkit, pilkut ja lainausmerkit mukaan lukien, siirretään. Merkkijonon viimeisiksi merkeiksi tulevat CR (vaunupalautus) ja LF (rivinsiirto).

```
60 INPUT LINE #2,S¤ 90 INPUT LINE #F, S¤
```

INTEGER

Aiheuttaa sen, että muuttuja-arvoja ja vakioita käsitellään normaalisti kokonaislukuina. Liukulukumuuttujat ja -vakiot merkitään pisteellä. INTEGER sijoitetaan ohjelmaan ensimmäisenä. Vrt. FLOAT.

KILL "... ... ..."

Poistaa ilmoitetun tiedoston levyltä. Tiedostotyyppi on annettava.

```
30 KILL "DR1:TESTI.BAC"
90 KILL A¤ + "TESTI.BAC"
```

LET ...

Sijoittaa arvon muuttujalle. LET voidaan jättää pois.

```
10 LET X = 5*S
20 T¤ = "LEENA"
```

NAME "... ... ..." AS "... ..."

Muuttaa tiedostonimen. Tiedostotyyppi on ilmoitettava.

```
90 NAME "DR1:VANHA.BAC" AS "UUSI.BAC"
```

NEXT ...

Lopettaa silmukan. Vrt. FOR ... TO ... STEP ...

NO EXTEND

Kumoaa EXTEND-käskyn vaikutuksen. Aiheuttaa sen, että ainoastaan muuttujanimiä, jotka koostuvat yhdestä kirjaimesta tai yhdestä kirjaimesta yhdistettynä numeroon 0—9, saa käyttää.

NO TRACE

Kumoaa TRACE-käskyn vaikutuksen. Vrt. TRACE.

ON ERROR GOTO

Aiheuttaa hypyn virheenkäsittelyrutiiniin ilmoitetulla rivinumerolla, mikäli ajon aikana sattuu virhe. Vrt. RESUME. ON ERROR GOTO ilman rivinumeroa kumoaa aiemman käskyn ON ERROR GOTO ...

```
10 ON ERROR GOTO 1000
```

ON ... GOSUB ... , ...

Aiheuttaa hypyn yhteen ilmoitetuista alirutiineista.

```
60 ON P GOSUB 100,200,300
```

(Jos P = 1, hypätään riville 100, jos P = 2 riville 200 jne.)

ON ... GOTO ... , ...

Aiheuttaa hypyn yhteen ilmoitetuista alirutiineista.

```
70 ON R GOTO 100,200, 300
```

(Jos R = 1, hypätään riville 100, jos R = 2 riville 200 jne.)

ON ... RESTORE ... , ...

Aiheuttaa sen, että yksi ilmoitetuista RESTORE-käskyistä suoritetaan. Vrt. RESTORE ...

```
90 ON S RESTORE 100, 200, 300
```

(Jos S = 1, RESTORE 100, jos S = 2 RESTORE 200 jne.)

KÄSKYT (jatkoa)

ON ... RESUME ...

Aiheuttaa sen, että yksi ilmoitetuista RESUME-käskyistä suoritetaan. Vrt. RESUME ...

```
80 ON T RESUME 100,200,300
```

(Jos T = 1 RESUME 100, jos T = 2 RESUME 200 jne.)

OPEN "... ... ..." AS FILE ...

Avaa ilmoitetun tiedoston lukemista varten ja antaa tiedostolle tiedostonumeron (0—255).

```
40 OPEN "DR1:OSOITE.TXT" AS FILE 3
60 OPEN "OSOITE.TXT" AS FILE 4
80 OPEN A¤ + ".TXT" AS FILE F
```

OPTION BASE

Määrää kenttämuuttujien pienimmäksi indeksiksi 0 tai 1. Vrt. DIM ... Käsky on sijoitettava ensimmäiseksi ohjelmaan.

POSIT # ... , ...

Osoittaa suorasaantitiedoston kohtaa, mistä lukeminen/kirjoitus tapahtuu. Vrt. funktiota POSIT (...) sekä käskyjä GET # ... , ... ja PUT # ... , ...

PREPARE "... ... ..." AS FILE ...

Luo ja avaa ilmoitetun tiedoston kirjoitusta varten ja sijoitaa tiedostolle tiedostonumeron (0—255).

```
30 PREPARE "DR1:OSOITE.TXT" AS FILE 3
50 PREPARE A¤ + ".TXT" AS FILE F
```

PRINT ... (vaihtoehtoisesti ; ...)

Tulostaa kuvaruutuun. Käskyä voi seurata lauseke sekä erikoisfunktiot TAB(S) ja CUR(R,S).

```
20 PRINT "RIITTA"
30 PRINT A,B,C    (Tulostus sarakkeisiin)
40 ;"ALA = "Y     (Tiheä tulostus)
```

PRINT # ... , ... (vaihtoehtoisesti ; # ... , ...)

Kirjoittaa ilmoitettuun tiedostoon.

```
30 PRINT #2, "SATUNNAISLUVUT"
```

PRINT USING "..." ...

Ilmoitetun lausekkeen tulostus esitetään halutussa muodossa. Ks. käyttöohje.

```
20 PRINT USING "+ # # # .# #" A;B
```

PUT # ... , ...

Ilmoitettuun tiedostoon viedään haluttu jonomuuttuja. Teidosto-osoitin päivittyy automaattisesti. Vrt. POSIT (...)

```
50 PUT #1,S¤
```

RANDOMIZE

Antaa RND-funktiolle satunnaisen alkuarvon.

READ ...

Sijoittaa muuttujille arvot DATA-lauseista.

```
10 READ B,C¤
```

REM ... (vaihtoehtoisesti ! ...)

Käytetään ohjelman kommenteissa. Kaksoispiste voidaan jättä pois !-merkin edellä.

```
80 REM *** ALIRUTIINI ***
90 A=5 ! alkuarvo
```

RESTORE ...

Aiheuttaa sen, että seuraava READ-lause lukee dataa alkaen tietystä DATA-lauseesta.

```
50 RESTORE
```

(Lukee dataa alkaen ensimmäisestä DATA-lauseesta.)

```
70 RESTORE 100
```

(Lukee dataa DATA-lauseista alkaen riviltä 100.)

RESUME ...

Aiheuttaa paluuhypyn virhekäsittelyrutiinista ilmoitetulle ohjelmariville. Jos rivinumero jätetään pois, hyppy tapahtuu takaisin sille ohjelmariville, jossa virhe tapahtui.

```
80 RESUME 10
90 RESUME
```

RETURN

Aiheuttaa paluuhypyn alirutiinista vastaavaa GOSUB-lausetta seuraavaan lauseeseen.

SINGLE

Aiheuttaa sen, että liukuluvut esitetään 7 numerolla. Kumoaa DOUBLE-käskyn vaikutuksen.

STOP

Lopettaa ohjelman ajon.

TRACE

Suoritettujen ohjelmarivien numerot tulostetaan kuvaruutuun tai tiedostoon (TRACE # ...) ajon aikana. Vrt. NO TRACE.

WHILE ...

WEND

Ohjelmajakso toistetaan niin kauan kuin annettu ehto on tosi.

```
10 WHILE X<10
20 PRINT X : X = X +1
30 WEND
```

MERKKIJONOFUNKTIOT

ASC (A¤) Vaihtoehto ASCII(A¤)

Antaa ilmoitetun jonon ensimmäisen merkin ASCII-arvon.

```
20 LET P=ASC(B¤)
```

CHR¤(...)

Antaa merkin, joka vastaa ilmoitettua ASCII-arvoa.

```
10 PRINT CHR¤(12)
20 P¤=CHR¤(P,R,S,T,U,V)
```

INSTR(S,A¤,B¤)

Etsii osajonoa B¤ jonossa A¤ alkaen kohdasta S. Funktio antaa ensimmäisen sellaisen paikan, jossa B¤ ensi kerran esiintyy jonossa A¤. Jos B¤ ei esiinny A¤:ssa, saadaan arvo 0.

```
30 R=INSTR(S,A¤,B¤)
```

LEFT¤(A¤,S) Vaihtoehto LEFT(A¤,S)

Antaa merkkijonon A¤ S ensimmäistä merkkiä.

```
30 D¤=LEFT¤(C¤,S)
```

LEN(A¤)

Antaa kokonaisluvun, joka vastaa ilmoitetun jonon pituutta.

```
20 X=LEN(A¤)
```

MID¤(A¤,S,T) Vaihtoehto MID(A¤,S,T)

Antaa T kpl merkkejä alkaen merkkipaikasta S ilmoitetussa merkkijonossa.

```
30 B¤=MID¤(A¤,S,T)
```

MID¤(A¤,S,T)="..."

Vaihtaa T kpl merkkejä alkaen merkkipaikasta S jonossa A¤ ilmoitettuun jonoon.

```
30 MID¤(A¤,T,4)="UUSI"
```

NUM¤(S)

Antaa numeerisen merkkijonon, jossa on samat merkit kuin, joka saadaan, kun ilmoitettu lukumuuttuja tulostetaan PRINT-lauseella. Blanko, joka on varattu plus-merkkiä varten, ei sisälly jonoon NUM¤(S).

```
10 B¤=NUM¤(S)
```

RIGHT¤(A¤,S) Vaihtoehto RIGHT (A¤,S)

Antaa ilmoitetun jonon kaikki merkit alkaen kohdasta S.

```
40 C¤=RIGHT¤(A¤,S)
```

SPACE¤(S)

Antaa merkkijonon, jossa on S kpl välilyöntejä.

```
20 B¤=SPACE¤(253)
```

STRING¤(S,T)

Antaa jonon, jossa on S kpl merkkejä, joiden ASCII-arvo on T.

```
30 C¤=STRING¤(20,45)
```

VAL(A¤)

Muuttaa ilmoitetun numeerisen jonon lukumuuttujaksi.

```
10 S=VAL(A¤)
20 S=VAL(".000000003")
```

+

Yhdistää merkkijonoja.

```
30 S¤=A¤+B¤+".BAS"
```

VIRHEILMOITUKSET

19 Useampia tiedostoja ei voi avata
20 Liian pitkä rivi (>160 merkkiä)
21 Tiedostoa ei löydy
32 Tiedostoa ei ole avattu
34 Tiedosto loppu
35 Tarkistussummavirhe luettaessa
36 Tarkistussummavirhe kirjoitettaessa
37 Virheellinen sektoriformaatti
38 Sektorinumero tiedoston ulkopuolella
39 Tiedosto kirjoitussuojattu
40 Tiedosto poistosuojattu
41 Levy täynnä
42 Levy ei valmis
43 Levy kirjoitussuojattu
44 Loogista tiedostoa ei ole avattu
45 Väärä looginen tiedostonumero
46 Väärä yksikkönumero
47 Virheellinen trap-numero
48 Virhe levykirjastossa
49 Väärä fyysinen tiedostonumero
51 Yksikkö varattu
52 Väärä toiminto yksiköllе
53 Funktionäppäin
54 IEC sekä lähetin että vastaanotin
55 IEC-vastaanotin ei toiminnassa
56 IEC-lähetin ei toiminnassa
57 Näppäimistömerkin ajoitusvirhe
58 Virheellinen merkki luettu
64 Virheellinen "NAME"-lause
68 Virheellinen aikamääritys
130 Liian suuri liukuluku
131 Indeksi sallitun alueen ulkopuolella
132 Liian suuri kokoluku
133 Virhe ASCII-aritmetiikkalauseessa
134 Indeksi merkkijonon ulkopuolella
135 Negatiivinen "SPACE¤", "STRING¤" tai "TAB"<1
136 Liian pitkä merkkijono
137 "DIM"iä ei saa lisätä
138 "ON"-lauseessa väärä arvo
139 "RETURN" ilman "GOSUB"ia
140 Virheellinen "RETURN"-muuttuja
141 Data loppu
142 Virheellinen argumentti funktiossa
143 Virheellinen "SYS"-funktio
144 Kielletty rivi
145 "FNEND" ilman "RETURN"ia
146 "PRINT USING"-virhe
147 Virheellistä dataa
148 Liian vähän dataa sisään
149 "RESTORE" ei osoita "DATA"-riville
150 Liian paljon dataa sisään
151 "RESUME" ilman virhettä
176 Graafinen piste näytön ulkopuolella
180 Tätä rivinumeroa ei löydy
181 Virheellinen hyppy funktioon
182 "NEXT" tai "WEND" puuttuu
183 "FOR" tai "WHILE" puuttuu
184 Väärä muuttuja "NEXT"-sanan jäljessä
185 "FOR"-silmukoita saman muuttujan kanssa
186 "FOR"-silmukka paikallismuuttujan kanssa kielletty
187 Funktiota ei ole määritelty
188 Useampia samannimisiä funktioita
189 Virheellinen funktio
190 Virheellinen indeksien määrä
191 Ei voi sijoittaa (funktiossa)
200 Yksikkö ei ole kytketty
201 Muisti täysi
202 "LIST"-suojattu ohjelma
203 Väärä ohjelmaformaatti
204 "MERGE" ei käy "BAC"-tiedostolla
205 "COMMON"-virhe
206 Käytä komentoa "RUN"
207 Ei voi jatkaa
208 Ei saa käyttää komentona
209 Komennossa väärää tietoa
210 Virheellinen luku
211 Tarkkuutta ei saa muuttaa
220 Syntaksivirhe
221 Kielletty merkki lauseen jäljessä
222 Tulee olla rivillä ensimmäisenä
223 Virheellinen argumenttien määrä tai laatu
224 Virheellinen lukujen ja jonojen yhdistelmä
225 Ei yksinkertainen muuttuja
226 Virheellinen lause "ON"-sanan jäljessä
227 ","-merkki puuttuu
228 "="-merkki puuttuu
229 ")"-merkki puuttuu
230 "AS FILE" puuttuu
231 "AS" puuttuu
232 "TO" puuttuu
233 Rivinumero puuttuu
234 Virheellinen muuttuja

Luxor Data, EMMDATA & WT-KONSULT
Art.n:o 66 79210-73

LAUSEKKEET JA OPERAATTORIT

Lauseke koostuu luvuista ja funktioista, joita erottavat operaattorit. Seuraavat operaattorit ovat olemassa:

PRIORITEETTI	OPERAATTORI	MERKITYS	ESIMERKKI
1	Ü (vaiht. * *)	eksponentti	AÜB
2	*	kertolasku	A*B
2	/	jakolasku	A/B
3	+	yhteenlasku	A+B
3	—	vähennyslasku	A—B
4	>	suurempi kuin	A>B
4	<	pienempi kuin	A<B
4	=	yhtä suuri kuin	A=B
4	>=	suurempi tai yhtä suuri kuin	A>=B
4	<=	pienempi tai yhtä suuri kuin	A<=B
4	<>	eri suuri kuin	A<>B

Vertailuoperaattorit, joiden prioriteetti on 4, antavat arvon TOSI (—1), jos lauseke on tosi, muutoin arvon EI-TOSI (0). Vrt. LOOGISET OPERAATTORIT.

LOOGISET OPERAATTORIT

Tässä kuvattavista loogisista operaattoreista on operaattorilla NOT korkein prioriteetti. OR- ja XOR-operaattoreilla on sama prioriteetti. Kaikilla loogisilla operaattoreilla on pienempi prioriteetti kuin muilla operaattoreilla.

NOT

EI. On tosi, jos operandi on ei-tosi.

```
10 IF NOT A<B THEN GOTO 20
```

AND

JA. On tosi, jos molemmat operandit ovat tosia.

```
20 IF A>B AND C=D THEN 30
```

OR

TAI. On tosi, jos vähintäin toinen operandeista on tosi.

```
30 IF A<B OR C=10 THEN 40
```

XOR

POISSULKEVA TAI. On tosi, jos vain toinen operandi on tosi, mutta eivät molemmat.

```
40 IF A=B XOR C=D THEN 50
```

IMP

IMPLISOI. A IMP B on ei-tosi vain, jos A on tosi ja B on ei-tosi.

```
50 IF A IMP B THEN 60
```

EQV

EKVIVALENSSI. On tosi, jos molemmat operandit ovat tosia tai molemmat ovat ei-tosia.

```
60 IF A=B EQV C=D THEN 70
```

Loogisia operaattoreita voidaan käyttää myös mielivaltaisten kokonaislukujen kanssa. Operaattorit toimivat silloin bitti bitiltä vastaavissa binaariluvuissa.

```
70 A%=B% AND 15%
```

ASCII-FUNKTIOT

ADD¤(A¤,B¤,T)

Summaa ilmoitetut numeeriset merkkijonot. Tulos pyöristetään ilmoitettuun (T) määrään oikeita desimaaleja. Numeeriset merkkijonot saavat sisältää numeroita (enintään 124 kpl), merkit + ja — sekä desimaalispisteen.

```
20 C¤ = ADD¤(A¤,B¤,T)
```

DIV¤(A¤,B¤,T)

Jakaa numeerisen merkkijonon (A¤) toisella (B¤). Vrt. ADD¤(A¤,B¤T).

```
30 C¤= DIV¤(A¤,B¤,T)
```

MUL¤(A¤,B¤,T)

Kertoo numeerisen merkkijonon (A¤) toisella (B¤). Vrt. ADC¤,B¤,T).

```
40 D¤ = MUL¤(A¤,B¤,T¤)
```

SUB¤(A¤,B¤,T)

Vähentää numeerisen merkkijonon (A¤) toisesta (B¤). Vrt. ADD¤(A¤,B¤,T).

```
50 C¤ = SUB¤(A¤,B¤,T)
```

COMP%(A¤,B¤)

Vertaa kahta numeerista jonoa. Funktio antaa seuraavat arvot vastaukseksi:

—1, jos A¤ < B¤
0, jos A¤ = B¤
+1, jos A¤ > B¤

```
10 P = COMP%(A¤,B¤)
```

MATEMAATTISET FUNKTIOT

ABS(X)	x:n itseisarvo, |x|
ATN(X)	x:n arctangentti
COS(X)	x:n kosini, x radiaaneina
EXP(X)	e^x
FIX(X)	x:n kokonaisosa, [x]
INT(X)	suurin kokonaisluku, joka on pienempi tai yhtä suuri kuin x
LOG(X)	e-logaritmi x:stä
LOG10(X)	10-logaritmi x:stä
MOD(X,Y)	jakolaskun X/Y jakojäännös
SGN(X)	—1, jos x<0. 0, jos x=0. 1, jos x>0
SIN(X)	sin x:stä, x radiaaneina
SQR(X)	neliöjuuri, x:stä, $\sqrt{x}$
TAN(X)	tangentti x:stä, x radiaaneina
PI	π, luku 3.14159 (3.141592653589793)
RND	esittää satunnaisluvun välillä 0—0.999999 (vaihtoehtoisesti välillä 0—0.9999999999999999)

ERITYISFUNKTIOT

CUR(R,S)

Siirtää kohdistimen (kursorin) riville R sarakkeeseen S kuvaruudulla. Esiintyy ainoastaan PRINT-lauseissa.

$0 \leq R \leq 23$ (rivit) $0 \leq K \leq 39/79$ (sarakkeet)

```
20 PRINT CUR (0,22)"OIKEALLA YLHÄÄLLÄ"
```

CVT ... ... (...)

Mahdollistaa merkkimuunnoksen (konvertoinnin) tiedoston lukemisessa/kirjoittamisessa. Seuraavat muunnokset ovat olemassa, R% on kokonaisluku ja T on liukuluku.

CVT %¤(R%)	R% konvertoidaan merkkijonoksi (2 tavua)
CVT ¤%(S¤)	S¤ konvertoidaan kokonaisluvuksi
CVT F¤(T)	T konvertoidaan merkkijonoksi (4/8 tavua)
CVT ¤F(S¤)	S¤ konvertoidaan liukuluvuksi

```
20 PUT #1,CVT F¤(T)
```

ERRCODE

Antaa tulokseksi viimeisimmän virhekoodin.

```
30 E9 = ERRCODE
```

HEX¤(...)
OCT¤(...)

Muuntaa annetun desimaaliluvun merkkijonomuotoiseksi heksadesimaaliluvuksi ja vastaavasti oktaaliluvuksi.

```
50 X¤ = HEX¤(128) : Y¤ = OCT¤(83)
```

POSIT(F)

Antaa tiedosto-osoittimen arvon tiedostossa, jonka numero on F.

```
50 PRINT POSIT (3)
```

SYS (...)

Antaa ilmoitetun järjestelmämuuttujan arvon. Seuraavat järjestelmämuuttujat ovat olemassa:

SYS(2)	Yksikön muistikoko (tavua)
SYS(3)	Nykyisen ohjelman koko (tavua)
SYS(4)	Käyttämättömän muistin koko (tavua)
SYS(5)	Näppäimistölippu
SYS(6)	Viimeisin merkki palautetaan näppäimistöpuskuriin
SYS(11)	Ohjelman alkuosoite
SYS(12)	Muuttujajuuri ensimmäiseen muuttujanimeen

TAB(T)

Siirtää kohdistimen rivin paikkaan T ($T \geq 1$). Esiintyy ainoastaan PRINT-lauseissa.

```
40 PRINT TAB(14);"**OTSAKE**"
```

TIME¤

Antaa sisäänrakennetun kellon päiväyksen ja kelloajan.

```
10 POKE —17,Y,M,D,H,M1,S ! asetus
20 PRINT TIME ¤
```

VAROOT(A¤)

Antaa osoitteen taulukkoon, jossa ovat tiedot muuttujasta A¤

```
30 PRINT VAROOT(A¤)
```

VARPTR(A¤)

Antaa alkuosoitteen alueelle, jonne muuttujan A¤ arvo on talletettu

MUISTINKÄSITTELY JA IN/OUT-PORTIT

CALL(A)

Kutsuu annetussa osoitteessa (A) olevaa konekielirutiinia. Suoritetun alirutiinin jälkeen CALL(A) = mikroprosessorin Z80 HL-rekisterissä oleva luku.

```
10 S = CALL(49800)
```

CALL(A,U)

Kutsuu ilmoitetussa osoitteessa (A) olevaa konekielirutiinia. Ennen alirutiinikutsua prosessorin DE-rekisteriin ladataan annettu ilmaus (U). Suoritetun alirutiinin jälkeen CALL(A,U) = HL-rekisterissä oleva luku.

```
20 C% = CALL(49800%,U%)
```

INP(P)

Noutaa tavun (8 bittiä) ilmoitetulta portilta (P).

```
30 C% = INP(34)
```

OUT P1,D1,P2,D2 . . .

Siirtää datan D1,D2 . . . ilmoitetulle portille (P).

```
40 OUT 58,32
```

PEEK(A)

Noutaa tavun (kahdeksan bittiä) ilmoitetusta muistiosoitteesta (A).

```
50 PRINT PEEK(49800)
```

PEEK2(A)

Noutaa kaksi tavua ilmoitetusta muistiosoitteesta A (vähiten merkitsevä tavu) ja seuraavasta osoitteesta (eniten merkitsevä tavu). Molemmat kahdeksan bitin ryhmät näytetään yhtenä kokonaislukuna.

```
40 B% = PEEK2(32000)
```

POKE A,D1,D2, . . .

Siirtää datan D1, D2, . . . ilmoitettuun muistipaikkaan alkaen osoitteesta (A). Data ja osoitteet annetaan desimaalilukuina.

```
60 POKE 65008,10,5,2
```

SWAP%(D)

Antaa D:n arvon tulkittuna kokonaisluvuksi (2 tavua), mutta ensimmäinen ja toinen tavu vaihdettuna keskenään.

```
70 B% = SWAP%(D%)
```

NÄPPÄIMISTÖ

[CE] Poistaa kirjoitetun rivin.

[CTRL] [L] Siirtää kohdistimen vasempaan yläkulmaan ja tyhjentää kuvaruudun.

[CTRL] [C] Keskeyttää meneillään olevan ohjelman ajon. Tätä voi seurata jokin seuraavista:

- [ ] Ohjelman ajo jatkuu (valinnainen näppäin).
- [CTRL] [S] Kullakin näppäimen painalluksella suoritetaan yksi ohjelmarivi.
- [CTRL] [C] Ohjelman ajo lopetetaan.

Funktionäppäinten toiminta on kuvattu käyttöohjeessa.

VÄRI JA GRAFIIKKA ABC 800C:LLE

Huomaa, että seuraavat erikoisilmaukset väriä ja grafiikkaa varten esiintyvät vain PRINT-lauseissa.

RED GRN YEL BLU MAG CYA WHT

Tulostaa tavalliset merkit värillisinä (merkkitila).

```
10 PRINT YEL "aurinko" RED "punainen" BLU
"meri"
```

GRED GGRN GYEL GBLU GMAG GCYA GWHT

Tulostaa grafiikkamerkit värillisinä (grafiikkatila).

```
20 PRINT GGRN "wä" GCYA "wä"
```

FLSH STDY

Antaa vilkkuvan ja vastaavasti normaalin tulostuksen.

```
30 PRINT FLSH YEL "tähti"
```

DBLE NRML

Tulostaa kaikinkertaisella ja vastaavasti normaalilla merkkikorkeudella.

```
40 PRINT RED DBLE "TAULUKKO"
```

NWBG BLBG

Muuttaa rivin taustavärin ja vastaavasti palauttaa sen normaaliksi (mustaksi). Haluttu taustaväri annetaan ennen NWBG-käskyä.

```
50 PRINT BLU NWBG YEL "AURINKO " BLBG
```

GSEP GCON

Antaa yhdistetyn ja vastaavasti erotetun grafiikan.

```
60 PRINT GMAG GSEP "777" GCON "777"
```

GHOL GREL

Antaa mahdollisuuden esittää ohjausmerkin paikalla edeltävän graafisen merkin.

```
80 PRINT GYEL "sss" GHOL GBLU "sssss"
```

HIDE

Antaa piilotekstin/-grafiikan.

```
70 PRINT "LAJITTELE" HIDE "NIMI"
```

TXPOINT X,Y,S

Sytyttää (S = 1 tai poisjätetty S) tai sammuttaa (S = 0) graafisen pisteen koordinaatissa X ja Y. Kohdan on oltava graafisessa tilassa.

$0 \leq X \leq 77$ $0 \leq Y \leq 77$

```
80 TXPOINT 77,71 ! OIKEALLA YLHÄÄLLÄ
```

TXPOINT (X,Y)

Antaa arvon TOSI (—1), jos kuvaruudun koordinaatti X,Y on sytytetty ja muutoin arvon EI-TOSI (0).

```
10 IF TXPOINT (X,Y) THEN TXPOINT X,Y,O
ELSE TXPOINT X,Y
```

VÄRI JA GRAFIIKKA ABC 800C:LLE (jatkoa)

SET DOT R,S

Sytyttää graafisen pisteen kuvaruudulla. Rivipaikan tulee olla graafisessa tilassa.

$0 \leq R \leq 71$ ja $2 \leq S \leq 79$

```
80 SET DOT 0,2 ! VASEMMALLA YLHÄÄLLÄ
```

CLR DOT R,S

Sammuttaa graafisen pisteen kuvaruudulla. Vrt. SETDOT R,S.

```
90 CLR DOT 71,79 ! OIKEALLA ALHAALLA
```

DOT (R,S)

Antaa arvon TOSI (—1), jos graafinen piste R,S kuvaruudulla on sytytetty ja muutoin arvon EI-TOSI (0).

```
10 IF DOT (R,S) THEN CLR DOT R,S
ELSE SET DOT R,S
```

ASCII-MERKIT MERKKITILASSA (M) JA GRAFIIKKATILASSA (G)

Näppäinkoodit merkki/grafiikkamuodossa (ASCII-taulukko)

A	M	G	A	M	G	A	M	G	A	M	G
32	Tyhjä		56	8		80	P	P	104	h	
33	!		57	9		81	Q	Q	105	i	
34	"		58	:		82	R	R	106	j	
35	#		59	;		83	S	S	107	k	
36	¤		60	<		84	T	T	108	l	
37	%		61	=		85	U	U	109	m	
38	&		62	>		86	V	V	110	n	
39	'		63	?		87	W	W	111	o	
40	(		64	É	É	88	X	X	112	p	
41	)		65	A	A	89	Y	Y	113	q	
42	*		66	B	B	90	Z	Z	114	r	
43	+		67	C	C	91	Ä	Ä	115	s	
44	,		68	D	D	92	Ö	Ö	116	t	
45	-		69	E	E	93	Å	Å	117	u	
46	.		70	F	F	94	Ü	Ü	118	v	
47	/		71	G	G	95	_	_	119	w	
48	0		72	H	H	96	é		120	x	
49	1		73	I	I	97	a		121	y	
50	2		74	J	J	98	b		122	z	
51	3		75	K	K	99	c		123	ä	
52	4		76	L	L	100	d		124	ö	
53	5		77	M	M	101	e		125	å	
54	6		78	N	N	102	f		126	ü	
55	7		79	O	O	103	g		127	■	■

ASCII-koodit (A) tulkittuina merkki- (M) ja grafiikkamuodossa (G).
Grafiikkamuoto kuuluu vain ABC 800 C-malliin.

CTRL ≧ Antaa merkin ■ (neliö)

KOMENNOT

¤BAS

Aiheuttaa paluun BASICiin. Vastaa RESET-näppäimen käyttöä. Vrt. BYE.

AUTO . . . , . . .

Aiheuttaa automaattisen rivinumeroinnin syöttämisen yhteydessä.

AUTO	(rivin:o 10.	Intervalli: 10)
AUTO 100	(rivin:o 100.	Intervalli: 10)
AUTO 500,50	(rivin:o 500.	Intervalli: 50)

BYE

Lopettaa BASIC-tulkin käytön. Sulkee kaikki tiedostot.

CLEAR

Nollaa kaikki muuttujat ja sulkee kaikki tiedostot.

CONTINUE (vaihtoehtoisesti CON)

Aiheuttaa sen, että ohjelman ajo jatkuu sen jälkeen, kun on suoritettu STOP-käsky tai on painettu kaksi kertaa CRTL-C. Vrt. NÄPPÄIMISTÖ.

ED . . .

Mahdollistaa ohjelmarivin muuttamisen ilman, että se täytyy kirjoittaa uudelleen. Näppäin [CTRL] kopioi yhden merkin ja vastaavasti näppäin [C] poistaa merkin.

ERASE . . .

Poistaa ilmoitetun ohjelman osan. Seuraavat muunnokset ovat olemassa:

ERASE 40—80	(poistaa rivit välillä 40—80)
ERASE —80	(poistaa rivit riviin 80 saakka)
ERASE 40—	(poistaa rivit rivistä 40 eteenpäin)

LIST . . .

Tulostaa ABC 800:ssa olevan ohjelman kuvaruutuun. Seuraavat muunnokset ovat olemassa.

LIST	(kaikki rivit)
LIST 30	(vain yksi rivi)
LIST 30—60	(rivit kahden rivinumeron välillä)
LIST 60—	(rivit alkaen tietystä rivinumerosta)

LIST PR:

Tulostaa ABC 800:ssa olevan ohjelman kirjoittimelle. PR:n vastine joissakin ohjelmissa on V24:

LIST PR:	(kaikki rivit)
LIST PR:,10—90	(rivit kahden rivinumeron välillä)

LIST . . . .

Siirtää ABC 800:ssa olevan ohjelman kasetille tai levylle kääntämättömässä muodossa (tekstimuodossa). Jos tiedostotyyppi jätetään ilmoittamatta, käytetään tyyppiä .BAS.

LIST OSAOHJE,20—40
LIST DR1:SORTTAUS
LIST CAS:TESTI.001

KOMENNOT

LOAD . . . . . .

Siirtää ohjelman, jonka nimi on ilmoitettu, kasetilta tai levyltä ABC 800:n muistiin. Poistaa olemassa olevan ohjelman ABC 800:sta.

```
LOAD BUDJETTI
LOAD DR1:KALKYYLI.001
LOAD CAS:      (siirtää ensimmäisen kasetilta tavattavan ohjelman)
```

MERGE . . . . . .

Siirtää ohjelman tekstimuodossa (talletettu komennolla LIST . . .) kasetilta tai levyltä ABC 800:aan poistamatta muistissa jo olevaa ohjelmaa. (Jos muistissa olevassa ohjelmassa on käytetty samoja rivinumeroita, ne korvautuvat ladattavasta ohjelmasta.)

NEW (vaihtoehtoisesti SCR)

Pyyhkii ohjelman ABC 800:n muistista ja sulkee kaikki tiedostot.

REN . . . , . . . , . . . — . . .
(vaihtoehtoisesti RENUMBER . . . , . . . , . . — . . .)

Ilmoitetuille riveille annetaan uudet rivinumerot. Seuraavat muunnokset ovat olemassa:

Seuraavat esimerkit koskevat kaikkia ohjelmarivejä:

```
REN          (ens. rivin:o  10. Intervalli: 10)
REN 100      (ens. rivin:o 100. Intervalli: 10)
REN 100,20   (ens. rivin:o 100. Intervalli: 20)
```

Seuraavat esimerkit koskevat ilmoitettua ohjelmaosuutta, jonka ensimmäinen rivinumero on 600 ja intervalli 10:

```
REN 600,10,500—800(rivin:ot 500—800)
REN 600,10,—800(rivin:ot riviin 800 saakka)
REN 600,10,500—(rivin:ot alkaen rivistä 500)
```

RUN

Ajaa ABC 800:n muistissa olevan ohjelman nollattuaan ensin kaikki muuttujat.

RUN . . . . . .

Vastaa komentoa LOAD . . . . . , jota seuraa komento RUN. Vrt. LOAD . . . . . Yksikkötyyppi sekä tiedostotyypit .BAC ja .BAS voidaan jättää pois.

SAVE . . . . . .

Siirtää ABC 800:n muistissa olevan ohjelman tiedostoksi kasetille tai levylle sisäisen koodin muodossa. Jos tiedostotyyppi jätetään pois, käytetään tyyppiä .BAC. Vrt. UNSAVE . . . . . .

```
SAVE SORTTAUS
SAVE DR1:SORTTAUS
SAVE CAS:TESTI.001
```

UNSAVE . . . . . .

Poistaa ilmoitetun tiedoston levyltä. Jos tiedostotyyppi jätetään pois, käytetään ensin tyyppiä .BAC ja sitten .BAS. Vrt. SAVE . . . . . .

PKA
61.39

2

2 VIN
BAS

PKA
61.39 2

LUNDGREN:
BASIC II

Kaupunginkirjasto

BASIC II-opas kuvaa mm. ruotsalaisissa ABC 800- ja FACIT DTC -tietokoneissa käytettävän, nykyaikaisen BASIC II -tyyppisen BASIC-muunnoksen ominaisuuksia.

BASIC II -opas selvittää helpolla tavalla, miten nykyaikaista tietokonejärjestelmää ohjelmoidaan ja käytetään. Samalla painotetaan sitä, miten tärkeätä on laatia ohjelmat vastaamaan käyttäjien tarpeita. Sen vuoksi kirjassa onkin laajasti selostettu tietokoneen ja sen käyttäjän kommunikointia helpottavia tekijöitä.

Kaikelle tietojenkäsittelylle oleellista tiedostojen hallintaa on käsitelty perusteellisesti. Kirjassa on paljon tietokoneille ABC 800 ja FACIT DTC tarkoitettuja esimerkkiohjelmia, jotka esittävät käytännön tietokonesovelluksia.

Kirjaa voidaan käyttää lukioiden atk-opetuksessa, ammattioppilaitoksissa ja vapaan sivistystyön piirissä. Perusteellisesti käsitellyt esimerkkiohjelmat samoin kuin lukuisat harjoitustehtävät ratkaisuineen helpottavat kirjan käyttöä yksinopiskelussa.

KOTKAN KAUPUNGINKIRJASTO
285 N 0029279

TUOTESELOSTE Kirjan pääasiallinen käyttöalue:

koulumuoto tai ala / koulusvuosi	10	11	12	13	14
lukio	■	■	■		
kauppa	■	■	■		
tekninen	■	■	■		
ammatilliset kurssikeskukset	■	■			
kansanopistot ja -korkeakoulut	■	■			
kansalaistyöväen- ja vapaaopistot	■	■			
keskiasteen kouluvuosi	1	2	3	4	5

Kirjaa voidaan soveltuvin osin käyttää myös muissa koulumuodoissa tai muilla aloilla, jatko- ja täydennyskoulutuksessa sekä itseopiskelussa.

merkkien selitys (Otavan kirjat) ■ tämä kirja ● muu valmis kirja □ tulossa

Tekijät
Jan Lundgren
Sören Thörnell
Suomentajat:
Fil. maist. Usko Lahti
LUK Harri Lehtimäki

Tavoite
Tietokoneohjelmoinnin perusteiden opetus ABC-800 ja FACIT DTC sovelluksin

Yhteydet
OTAVA. KOULUOSASTO
Uudenmaankatu 10. 00120 Helsinki 12
Puh. 90-647 022

otava K61
Tilno B935
ISBN 951-1-07316-8